

Implementació d'un sistema de gestió de
cursos (Course Management) al Campus
Virtual de la UdL

Mireia Calzada Doladé

Co-directors:

Ramón Bejar Torres i David Barroso Iglesias

Juny 2014

Índex

Introducció	1
1 Descripció del projecte	3
1.1 Motivació	4
1.2 Objectius	5
1.3 Estructura del document	6
2 El model acadèmic de la UdL	7
2.1 Planificació d'una assignatura a la UdL	11
3 El Campus Virtual de la UdL	15
3.1 Conceptes sobre Sakai	16
3.1.1 Espais	18
3.1.2 Pàgines	19
3.1.3 Eines	19
3.1.4 Realms	19
3.1.5 Rols	19
3.1.6 Permisos	20
3.1.7 Proveïdors	21
3.1.8 Creació d'espais	22

4	Course Management: El model teòric	25
4.1	Objectius	26
4.2	Conceptes bàsics	27
4.2.1	CourseSet	28
4.2.2	CanonicalCourses	29
4.2.3	CourseOffering	29
4.2.4	Section	30
4.2.5	EnrollmentSet	31
4.2.6	Enrollment	31
4.2.7	Membership	31
4.2.8	Academic Session	32
4.3	Casos d'ús	32
4.4	Integració amb Course Management	35
4.4.1	Clients de la API CM	35
5	Course Management: La implementació a la UdL	39
5.1	Anàlisi	40
5.2	Disseny	45
5.2.1	Matching de conceptes	45
5.2.2	Consideracions per la implementació a la UdL	49
5.3	Implementació	54
5.3.1	Implementació de l'API del Course Management	54
5.3.2	Implementació del procés de sincronització	56
5.3.3	Jobs de Quartz	68
5.4	Test	70
5.4.1	Pla de test	71

<i>ÍNDIX</i>	v
6 Metodologia i eines emprades	73
6.1 Metodologia de desenvolupament	77
6.2 Eines emprades	80
7 Conclusions i treballs futurs	91
A Detall dels fitxers inclosos en la implementació	97
Bibliografía	102

Índex de figures

2.1	Model universitari abans del procés de Bolonya	9
2.2	Model univesitari actual	9
2.3	Exemple bàsic de graf de pla d'estudis	11
2.4	Exemple genèric de planificació d'una assignatura	13
3.1	Exemple d'espais, pàgines i eines	16
3.2	Usuaris i rols	17
3.3	Model conceptual de seguretat en Sakai	21
4.1	Relació jeràrquica entre els objectes del course management . . .	27
4.2	Diagrama de classes del Course Management	28
4.3	Exemple de la relació jeràrquica entre CourseSets i Canonical- Courses	29
4.4	Exemple de la relació jeràrquica entre CourseOfferings i Sections	30
4.5	Cas d'ús: Patró simple	33
4.6	Cas d'ús: Patró amb Sections i Enrollments al CourseOffering . .	33
4.7	Cas d'ús: Patró amb Sections i Enrollments a les mateixes Sections	34
4.8	Exemple d'eina orla	36
4.9	Exemple d'eina Informació de l'espai	37
4.10	Pantalla inicial en la creació d'un nou espai mitjançant l'eina Con- figuració de l'espai.	37

4.11	Llistat de sections planificades per a un professor que poden ser afegides a l'espai creat.	38
5.1	Relació jeràrquica de conceptes	45
5.2	Cas d'ús 1 UdL	46
5.3	Cas d'ús 2 UdL	47
5.4	Cas d'ús 3 UdL	48
5.5	Cas d'ús 4 UdL	49
5.6	Entitat relació resultant de les classes del Course Management en la implementació a la UdL	54
5.7	Model de dades que utilitzarem per la integració del Course Management a la UdL	55
5.8	Procés de sincronització	59
5.9	Fase I: Còpia de les dades per tenir referència	61
5.10	Fase II: Còpia de les dades de UXXI a SAKAI	62
5.11	Fase III: Tractament de baixes de participants i contenedors	64
5.12	Fase IV: Tractament de les altes de participants i contenedors	65
5.13	Fase V: Actualitza els estats de diferents elements	67
5.14	Exemple de test de funcionalitat	72
6.1	Metodologia tradicional vs Metodologia àgil	75
6.2	Definició inicial de les iteracions del projecte	76
6.3	Diagrama de flux integració de codi	79
6.4	Revisió de codi amb Subversion	80
6.5	Exemple de tasques planificades per als diferents entorns amb Jenkins	81
6.6	Exemple de tasques definides per al Projecte Course Management	82
6.7	Tasca concreta de Jira	83

6.8	Exemple de documentació escrita al Confluence	84
6.9	Finestres de treball en Eclipse	85
6.10	Finestres de treball del Sqldeveloper	86
6.11	Finestres de treball d'Spoon	87
6.12	Repositori Subversion on desem ETLs en entorn devel	88
6.13	Repositori Subversion on desem ETLs en entorn prod	88
6.14	Ús eines en el desenvolupament	89
A.1	Contingut resultant dels fitxers de la implementació	98
A.2	Detall fitxers resultants de la implementació dels treballs de Quartz	99
A.3	Estructura general de directoris dels fitxers ETLs	100
A.4	Contingut dels directoris d'ALTES i BAIXES	101
A.5	Job i transformacions encarregades de fer una còpia de les dades actuals CourseManagement i de tornar a copiar les dades des del sistema origen	102
A.6	Job i transformacions encarregades de tractar estats	102

Introducció

La Universitat de Lleida (en endavant UdL) és una institució amb una llarga tradició universitària que aposta per una recerca i una docència de qualitat.

Actualment, les universitats estan en constant avaluació. D'aquestes avaluacions se'n deriven classificacions de tot tipus que fan que la competència que hi ha entre les diferents universitats es converteixi en un recurs per a millorar-les.

La competitivitat generada entre les diferents universitats es tradueix en una millora contínua en la cerca de l'excel·lència en tots els aspectes del funcionament de la pròpia universitat, ja sigui com administració pública, en la qualitat de la docència i en la repercussió de la recerca que s'hi desenvolupa.

Cal convertir les universitats en opcions interessants per a nous alumnes i també per a professionals de primera línia que puguin dur a terme la seva recerca amb les millors garanties d'èxit, de manera que aquestes avaluacions també serveixen per fer evident que cal evolucionar les universitats cap entorns tecnològicament més avançats.

En aquest sentit, la UdL cita en la Declaració inicial dels seus Estatuts:

La universitat de Lleida vol proporcionar una formació de qualitat a les persones i el desenvolupament, com a valors fonamentals, de la qualitat en totes les seues activitats, la diferenciació en la docència i la recerca i el millor ús dels recursos disponibles.

Aquesta intenció es consolida en el pla estratègic 2013-2016.

El pla estratègic 2013-2016 formula com a objectiu: "Assolir la compatibilitat de les diferents plataformes informàtiques i de comunicació i de les diverses bases de dades de la UdL i millorar les plataformes actuals".

Dins aquest marc, les TIC (tecnologies de la informació i la comunicació) són una eina imprescindible, doncs permeten millorar totes les tasques que es desenvolupen en la universitat (administració, docència virtual, ...)

La finalitat d'aquest projecte és fer un pas endavant en aquesta direcció, oferint al professorat de la UdL eines per a la millor gestió de la seva docència dins el marc del Campus Virtual.

Capítol 1

Descripció del projecte

Aquest projecte pretén aconseguir un Campus Virtual més flexible i més obert, on el professorat pugui fer una gestió més acurada i específica dels seus espais de docència, a més de poder crear-se d'altres espais organitzant els grups de matrícula de la forma que més li convingui.

La idea és canviar la manera com s'aprovisiona actualment el Campus Virtual amb les dades de gestió acadèmica per tal d'aconseguir un campus on el professorat tingui totes les eines i dades disponibles per a que les pugui utilitzar de la manera que cregui més adient, i no d'una manera predefinida i tancada com es feia fins ara, on el professorat es trobava un espai per cada assignatura que impartia, però no en podia modificar gairebé res.

A banda d'aquest objectiu fonamental, aquest treball facilitarà en gran mesura les tasques realitzades per l'equip que gestiona el Campus Virtual en els processos de gestió d'espais de docència i del tractament dels seus participants en sincronització amb els sistemes origen.

1.1 Motivació

El model que ofereix actualment el Campus Virtual al professorat és un model tancat restrictiu, amb poc marge d'adaptació a les seves necessitats. Els espais de docència que se'ls proporciona contenen un llistat d'usuaris tancat que el mateix professorat de l'assignatura no pot tractar.

Des del punt de vista tècnic el procés de sincronització de totes les dades necessàries amb el sistema origen per a poder crear els espais de cada assignatura, els grups de matrícula i assignar-hi els participants ha esdevingut de gran complexitat, doncs s'ha hagut de tractar tots els casos que s'han anat produint (canvis de grup, canvis de professorat, ...) generant finalment un procés que requereix molt manteniment i molta intervenció per part del personal de suport.

Tot això es degut a que els espais d'assignatura al Campus Virtual (Sakai) no treuen tot el profit que n'haurien de treure de les dades acadèmiques per a poder representar estructures de cursos complexes, i a més, actualment no hi ha una forma automatitzada per crear dins de cada espai els diversos grups d'alumnes d'una assignatura.

Aquesta connexió deficient entre el Campus Virtual i els sistemes origen de les dades, com s'ha comentat, té un impacte negatiu en tota la comunitat universitària que en fa ús.

En resum, ens trobem davant un model tancat i limitat, en el que ni professors ni alumnes poden crear-se els seus propis espais, en el que les llistes d'usuaris són tancades i els professors no les poden gestionar. I a més l'esforç necessari per part del personal de suport del Campus Virtual per mantenir aquests espais i les dades que hi inclouen, és desproporcionat.

Queda evident, que cal evolucionar cap un model més obert, que doni més flexibilitat a professors i alumnes en la gestió dels seus propis espais en el Campus Virtual, doncs aquests col·lectius són en definitiva, els nostres clients.

1.2 Objectius

L'objectiu fonamental d'aquest projecte és obtenir un Campus Virtual més flexible per al professorat que els permeti gestionar els seus propis espais adaptant-los a les seves necessitats i preferències. Per aconseguir-ho, cal evolucionar la integració entre el Campus Virtual i els sistemes origen, fent-la més completa.

Aquesta nova sincronització ha de permetre definir les relacions entre les estructures de cursos i els espais al Campus Virtual, aconseguint així que aquests espais puguin ser manegats més fàcilment tant pels administradors del sistema com pel professorat, i això repercutirà també en un benefici directe en els estudiants.

Sakai, en la seva versió 2.3 va introduir una nova API amb el propòsit de definir un model que representés l'organització de cursos en institucions d'educació superior.

La implementació i integració d'aquesta API de Sakai al Campus Virtual de la UdL, ens permetrà aconseguir l'objectiu proposat, obtenint:

- Sincronització total amb el sistema de dades origen, que resulti pràcticament transparent al personal que gestiona el Campus Virtual.
- Més llibertat en la gestió i configuració dels espais d'assignatura per part dels professors, de manera que puguin organitzar els seus espais d'assignatura juntament amb els seus grups de la manera que més els convingui. Tant dels espais d'assignatura creats automàticament a l'inici de curs, com dels espais propis que podrà crear a partir d'ara el professorat.
- Més granularitat en l'assignació de participants als espais. Fins ara els participants s'assignaven directament als espais del campus, i era l'equip tècnic qui s'havia de preocupar de llegir els sistemes origen per assignar-los als

espais corresponents. Amb la nova implementació, les dades s'estructuraran de tal manera que serà el propi campus qui llegirà les dades del sistema origen i mostrarà a cada participant els espais que li pertocuin.

1.3 Estructura del document

Aquest document s'estructura en set capítols. En aquest primer capítol faig una descripció del propi projecte, la motivació que l'ha originat i els objectius del mateix.

Per tal de fer la lectura del document més entenedora, s'ha considerat oportú incloure dos capítols explicant d'una banda el model acadèmic universitari i en concret el de la UdL i, d'altra banda, una petita introducció al Campus Virtual (Sakai) exposant una sèrie de conceptes previs que s'utilitzaran al llarg del projecte.

El quart capítol descriu el model teòric de la solució que adoptarem en el cas que plantegem a la UdL, mentre que el cinquè capítol aprofundeix en l'explicació de la implementació concreta al Campus Virtual de la UdL.

En el sisè capítol s'explica quines eines de treball hem utilitzat i quina metodologia s'ha seguit a l'hora de implementar i implantar la solució.

Finalment, en el darrer capítol he resumit les conclusions extretes de la realització d'aquest projecte i es proposen possibles treballs futurs relacionats.

Capítol 2

El model acadèmic de la UdL

La Declaració de Bolonya (1999), subscripta per 45 estats europeus, va sentar les bases per adaptar cada sistema universitari a un nou marc d'educació comú que es coneix com Espai Europeu d'Educació Superior (EEES), organitzat conforme a certs principis: qualitat, mobilitat, diversitat, competitivitat, i orientat a la consecució entre d'altres de dos objectius estratègics: l'increment de la ocupació a la Unió Europea i la conversió del sistema Europeu de Formació Superior en un pol d'atracció per a estudiants i professors d'altres parts del món.

Amb l'adopció d'aquest nou sistema d'educació (EEES) es pretén aconseguir els següents objectius:

- L'adopció d'un sistema fàcilment comparable de titulacions, mitjançant la implantació, entre d'altres qüestions, d'un Suplement Europeu al títol.
- L'adopció d'un sistema basat, fonamentalment, en dos cicles principals: Graus i Màsters.
- L'establiment d'un sistema de crèdits, com el ECTS (European Transfer Credit System).

- La promoció de la mobilitat d'estudiants, professors i personal administratiu de les universitats i altres institucions d'ensenyament superior.
- La promoció de la cooperació Europea per tal d'assegurar un nivell de qualitat en el desenvolupament de criteris i metodologies comparables.

El que pretén tot aquest procés no és implantar un sistema únic a tot Europa, sinó facilitar la comparació de títols, establir objectius comuns i reforçar tot allò que és necessari per fer les universitats europees més atractives i competitives internacionalment.

Com a conseqüència de les modificacions establertes en el Tractat de Bolonya, s'estableix la formació universitària de la següent manera:

- Estudis de Grau: Amb l'objectiu d'oferir una formació de caràcter general dirigida a la preparació de l'exercici d'activitats professionals. Els plans d'estudi de Grau tenen una durada de 240 crèdits.
- Estudis de Màster: Tenen la finalitat que l'estudiant obtingui una formació avançada, de caràcter especialitzat o multidisciplinari, orientat a la especialització acadèmica o professional, o bé, promoure la iniciació a tasques investigadores. Els plans d'estudi de Màster tenen una durada d'entre 60 i 120 crèdits.
- Estudis de Doctorat: Els estudis de doctorat consten d'un període de formació i d'un període d'investigació. Tenen com a finalitat la formació avançada de l'estudiant en les tècniques d'investigació. Per poder accedir al període d'investigació cal estar en possessió d'un títol oficial de màster o bé d'algun altre títol expedit per una institució d'educació superior de l'EEES.

A continuació es mostra gràficament el canvi de la organització de les titulacions abans i després de Bolonya:

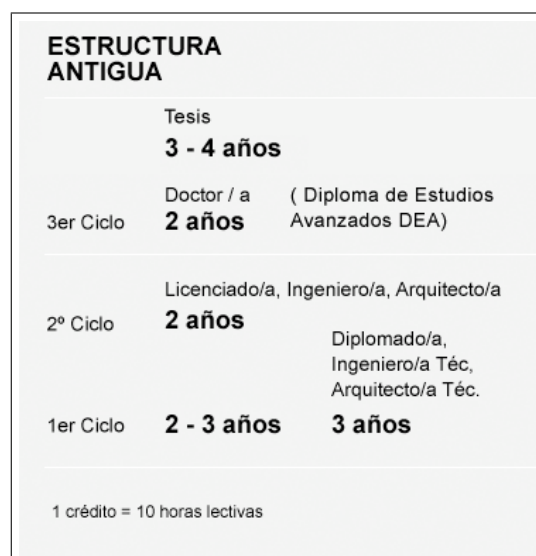


Figura 2.1: Model universitari abans del procés de Bolonya

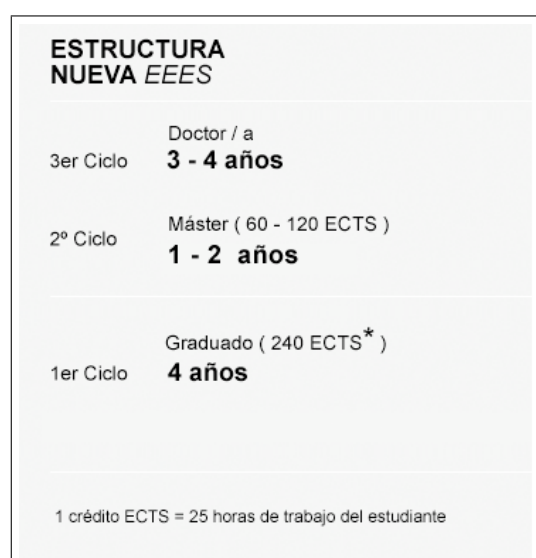


Figura 2.2: Model univesitari actual

La UdL va crear la Unitat de Planificació Docent (UPD) per a donar suport a la comunitat universitària en el procés d'adaptació de les titulacions al nou marc d'estudis superiors d'acord amb la normativa vigent i la política del Vicerectorat de Docència de la UdL.

El procediment per poder ofertar una titulació passa per molts estaments burocràtics, però quan finalment s'aprova des de la Generalitat, el resultat és una definició d'un pla d'estudis que cal oferir als alumnes per a que puguin matricular-s'hi.

Aquesta definició de pla d'estudis es realitza en forma de graf. Un graf és un diagrama format per nodes i arcs, que a la vegada estan compostos per blocs i sub-blocs d'assignatures o matèries. En un graf els nodes representen cursos, cicles o situacions intermitges en les que es pot trobar un alumne en funció de la naturalesa del pla d'estudis. Un arc uneix un node origen amb un node destí, associat a cada arc hi ha unes condicions relacionades amb les assignatures. Aquestes relacions permeten definir assignatures que cal cursar obligatòriament o bé un conjunt d'assignatures de les quals cal superar un nombre de crèdits (optatives) per a poder assolir el següent node.

La correcta definició del pla d'estudis garanteix el seguiment de cada alumne i la seva trajectòria acadèmica des de l'inici dels estudis fins l'acabament.

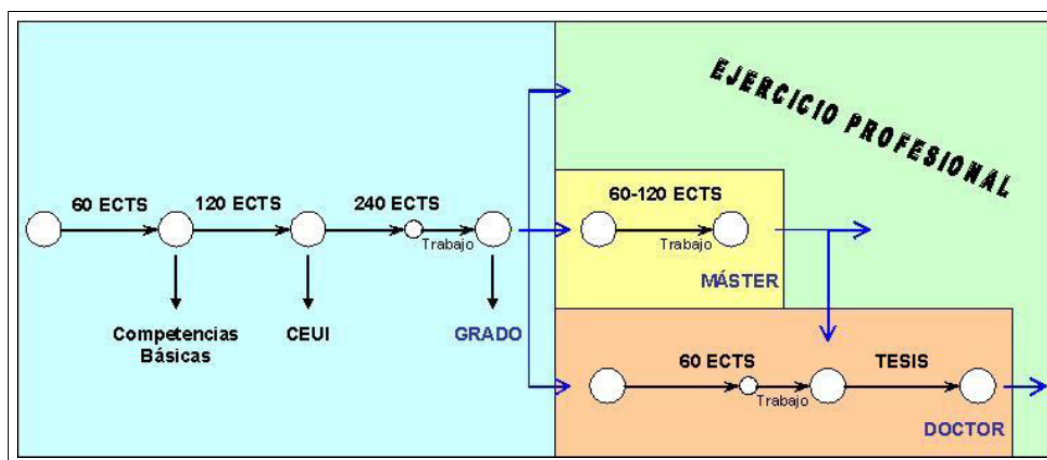


Figura 2.3: Exemple bàsic de graf de pla d'estudis

Un cop aprovats aquests plans d'estudi, el Servei de Gestió Acadèmica els introdueix a l'aplicació informàtica que ho gestiona mitjançant la representació gràfica dels diferents estadis pels que ha de passar un estudiant fins aconseguir l'obtenció del títol.

Totes les matèries definides en els diferents arcs del pla d'estudis, cal que estiguin degudament planificades en la mateixa aplicació de gestió acadèmica per tal que els alumnes puguin matricular-s'hi. Aquesta tasca la realitzen les secretaries de planificació docent de cada facultat o centre any rere any.

2.1 Planificació d'una assignatura a la UdL

La secretaria de planificació de cada centre és la responsable d'introduir l'oferta anual a l'aplicació de gestió acadèmica, un cop els departaments ja l'han aprovat.

L'oferta consisteix en un seguit d'assignatures que s'oferiran als estudiants per a que les puguin cursar.

La planificació concreta de cada assignatura, ha de preveure quines activitats formaran part de l'assignatura (pràctiques, teòriques, laboratori, tutories, etc.) i

especificar quina d'aquestes activitats serà la principal. A més, per cadascuna d'aquestes activitats cal que en facin una previsió, basada en l'experiència d'anys anteriors, sobre el nombre d'alumnes que es preveu que es matriculin.

De manera que per cada activitat, en funció del nombre d'alumnes previstos i de les capacitats de les aules, es defineixen els grups necessaris.

La planificació també ha de plasmar els professors que imparteixen cadascun d'aquests grups, amb el nombre de crèdits assignats a cadascun dels professors i identificant el professor que serà el responsable de l'assignatura.

Un cop introduïda la planificació de totes les assignatures ofertades, de totes les titulacions de tots els centres de la UdL, està tot llest per a poder començar les matrícules dels estudiants.

Els estudiants es matriculen a un grup concret de l'activitat principal. La resta de la planificació de l'assignatura, en el cas de la UdL, és bàsicament informatiu i orientatiu.

A continuació es mostra de forma gràfica la relació de tots els conceptes involucrats en la planificació d'una assignatura a l'aplicació de gestió acadèmica, doncs dins l'àmbit d'aquest projecte, és important tenir clars aquests conceptes ja que més endavant caldrà relacionar-los amb una altra estructura jeràrquica que proposa la gestió de cursos de Sakai.

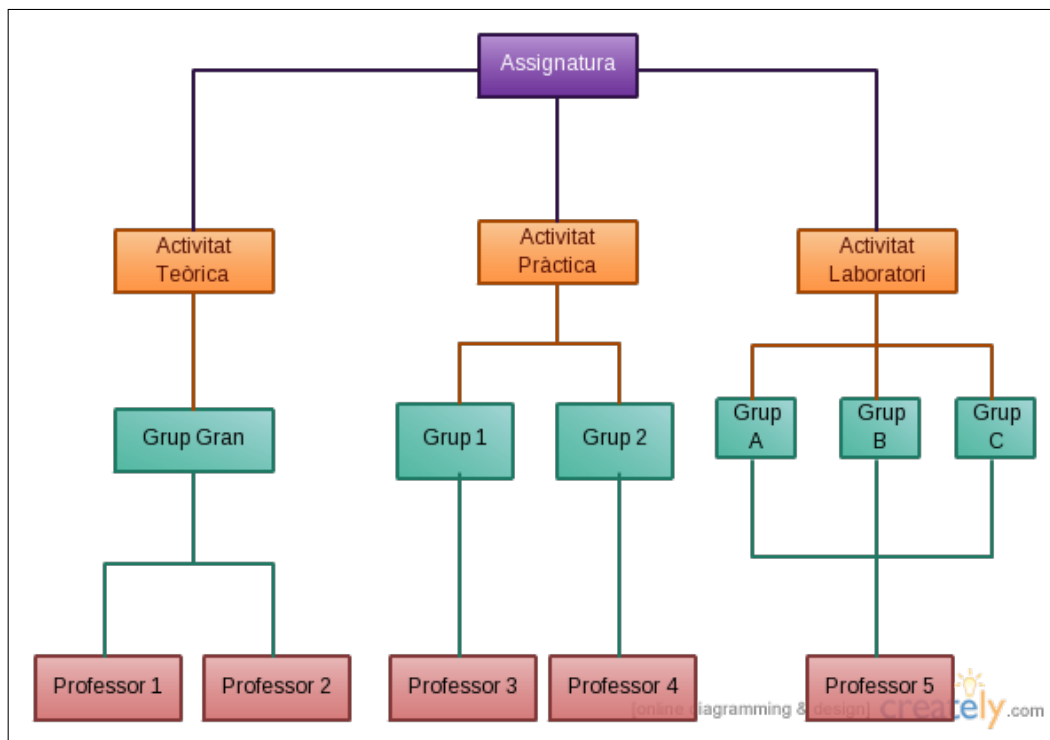


Figura 2.4: Exemple genèric de planificació d'una assignatura

Aquesta planificació d'una assignatura cal realitzar-la per totes les assignatures ofertes en cada titulació.

Capítol 3

El Campus Virtual de la UdL

Qualsevol institució docent disposa avui dia d'un sistema de gestió de l'aprenentatge (LMS), doncs com en tots els àmbits, la docència també ha evolucionat cap a l'ús de les noves tecnologies.

La Universitat de Lleida, l'any 2004 va optar per l'adaptació i ús de la plataforma de software lliure Sakai com a sistema de gestió de l'aprenentatge.

Sakai està avalat per les millors universitats del món (Cambridge, Stanford, Yale, Oxford, etc) i recolzat per una àmplia comunitat d'experts. Al ser una plataforma Open Source, està en constant ampliació i millora, cobrint d'aquesta manera les noves necessitats que sorgeixen en el món de la docència virtual.

Sakai ofereix una extensa llista d'eines que cobreixen gran part de les necessitats de formació on-line d'una organització. Aquest ventall d'eines permeten usar-lo com a reforç a la formació presencial, semi-presencial y on-line pròpiament dita. Permet també crear espais de treball col·laboratiu, pràctics per a la investigació o bé pel desenvolupament de projectes.

3.1 Conceptes sobre Sakai

La unitat bàsica en què s'estructura la informació a Sakai és l'espai.

La finalitat de dividir la informació en espais és justament per poder acollir en cadascun d'aquests espais continguts i participants amb interessos comuns, ja siguin assignatures, projectes col·laboratius, ...

Dins de cada espai, la informació s'organitza en pàgines. I a cada pàgina s'hi pot incloure una o més eines.

Les eines són les que mostren continguts, i les pàgines i espais són els contenidors d'aquests continguts.

The screenshot shows the Sakai Campus Virtual interface. At the top, there is a header with the University of Lleida logo and navigation links. Below the header, there is a sidebar on the left labeled 'Pàgines' (Pages) containing a list of navigation items: Inici, Programació, Recursos, Activitats, Tests, Missatges, Debat, Sala de xat, Agenda, Anuncis, Espai compartit, Qualificacions, Users, Seccions, Informació de l'espai, and Ajuda. The main content area is labeled 'Espais' (Spaces) and contains several sections: 'Agenda' with a calendar for June 2014, 'Anuncis recents' (Recent announcements), 'Gestió de presència' (Attendance management), 'Notificacions de comunicacions' (Communication notifications), and 'Missatges de xat recents' (Recent chat messages). The interface is designed to organize content into pages and spaces, with tools (eines) used to display content within these structures.

Figura 3.1: Exemple d'espais, pàgines i eines

Els espais són també la unitat bàsica de seguretat. A cada espai s'ha de definir els usuaris que hi tenen accés i quin rol desenvoluparan a l'espai. Això es pot fer manualment, assignant la tupla usuari-rol que pertany a cada espai, o bé mitjançant un proveïdor. Els proveïdors ens permeten indicar, per un espai o grup concret, qui serà l'encarregat d'aprovisionar d'usuaris i rols aquell espai o grup.

Els rols permeten agrupar permisos per a realitzar determinades accions.

⊙ Perfils d'accés ⓘ

Elimina el perfil d'accés Afegeix un rol Afegeix un usuari Desa com a

Edició: Perfil d'accés /site/cexemple

Utilitzat per Site: Curs exemple (cexemple) Created: 18/10/2006 10:32 by Administrador (admin) <p>Plantilla curs 06/07</p> Reviseu i modifiqueu la informació d'aquests permisos.

Perfil d'accés

Identificador del perfil d'accés: /site/cexemple

Identificador del proveïdor:

Rol d'administrador:

Acaba l'edició del perfil d'accés

Desa Cancel·la

Usuaris

Aquests són els rols concedits a usuaris individuals en aquest perfil d'accés. Feu clic sobre l'identificador per modificar-ne les propietats.

Identificador de l'usuari	Rol	actiu	De proveïdor
alex	Instructor	✓	
david	Instructor	✓	
h4371719	Student	✓	
m4374805	Student	✓	

Acaba l'edició del perfil d'accés

Desa Cancel·la

Rols

Aquests són els rols definits en aquest perfil d'accés. Feu clic en un identificador per modificar-ne les propietats.

Identificador del rol	Descripció
Instructor	Can read, revise, delete and add both content and participants to a site.
Student	Can read content, and add content to a site where appropriate.
Teaching Assistant	Can read, add, and revise most content in their sections.
convidat	Can read content, and add content to a site where appropriate.
maintain	

Figura 3.2: Usuaris i rols

A tall d'exemple, en aquest curs de prova veiem els següents rols:

- **Maintain:** Podrà realitzar qualsevol acció a l'espai
- **Convidat:** Únicament podrà accedir a l'espai, però no podrà realitzar cap acció per a modificar continguts
- **Instructor:** Pot tenir permís de lectura i escriptura en la majoria d'eines
- **Student:** Només pot accedir als continguts que publica l'Instructor.

En la imatge es pot comprovar que els participants han estat afegits manualment, doncs en la columna 'De proveïdor' no està marcada.

Els participants, dins els espais es poden organitzar en grups. Existeixen dos tipus de grups, els que es creen els professors per dividir els alumnes a l'hora de fer alguna activitat, i els grups predefinits que llegeix el campus del sistema origen. Aquests últims mostren els alumnes organitzats de la mateixa manera que al sistema origen (matrícula).

Algunes definicions importants sobre l'entorn de Sakai són els següents:

3.1.1 Espais

Un espai és un punt de reunió de participants, un lloc de treball que alberga un contingut relacionat amb ell. Pot ser públic, de manera que qualsevol usuari no autenticat pot accedir-hi, o privat.

Com ja hem comentat, els espais poden tenir diferent finalitat, com per exemple, per una assignatura, projecte col·laboratiu, etc., i per això existeixen diferents tipus d'espais:

- El meu espai de treball: espai personal associat a cada usuari, des d'aquí cada usuari pot configurar la seva visió del campus.
- Course: Espais associats a assignatures.
- Project: Espais col·laboratius.

A part d'aquests tipus d'espais, se'n poden definir d'altres, indicant quines eines contindran, tipus d'usuaris, permisos, etc.

3.1.2 Pàgines

Les pàgines són contenidors d'eines. És la manera que usa Sakai per organitzar les diferents eines que s'afegeixen a un espai.

Cada pàgina pot contenir una o varies eines i el títol tant de les pàgines com de les eines es pot personalitzar.

3.1.3 Eines

Una eina proporciona una funcionalitat que és usada des d'un espai a través d'una interfície web.

Una eina defineix una sèrie de permisos, de manera que els usuaris que tenen certs permisos poden accedir a certa informació, o bé realitzar una acció concreta.

Les eines a més, poden ser "*group aware*", això significa que l'eina en sí pot tenir diferent comportament per a cada grup de l'espai. Per exemple, els usuaris del grup A, poden veure els recursos d'una carpeta A, però no els que hi ha publicats a la carpeta B.

3.1.4 Realms

Els realms són utilitzats per a definir quines accions pot fer cada rol en un determinat espai. A banda de definir aquests permisos, també contenen per a cada rol una llista dels usuaris que tenen aquell rol en aquell espai.

3.1.5 Rols

Com hem vist, els usuaris que pertanyen a un espai, tenen un rol associat.

Aquests rols s'identifiquen mitjançant un nom i s'associen a un conjunt de permisos que utilitzen les eines per determinar l'accés a certes accions o informació de la eina.

Cada permís té dos possibles valors (actiu o inactiu)

Els rols que hi ha per defecte a Sakai, són:

- *.auth/.anon*: Diferència entre usuaris autenticats i usuaris que no han iniciat sessió.
- *access/maintain*: El rol *maintain* és el rol de manteniment, és l'administrador de l'espai, mentre que el rol *access* serveix per accedir a un espai, però sense cap permís per configurar-lo.
- *Instructor/Student/Teaching Assistant*: Aquests són els rols que s'utilitzen per a espais docents (tipus *course*). El rol de manteniment en aquests espais és l'*Instructor*

3.1.6 Permisos

Un permís serveix determinar l'accés a certa acció o poder visualitzar certa informació.

Per exemple, el permís *annc.newpermission* controla quins usuaris poden crear anuncis. Si un participant d'un espai té assignat un rol que tingui aquest permís, podrà crear anuncis.

En l'accés a una eina, el sistema comprova els permisos en el realm de l'espai, i si la eina és *group aware*, es comprovarà els permisos en la unió del realm de l'espai i el realm del grup.

En eines que utilitzen recursos, es comprovarà en primer lloc els permisos del realm del site, els permisos del realm del grup i finalment els permisos que es té en el recurs sobre el qual es vol realitzar l'acció.

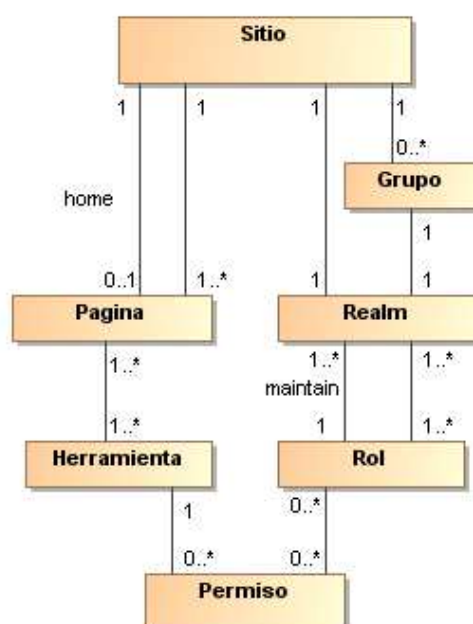


Figura 3.3: Model conceptual de seguretat en Sakai

3.1.7 Proveïdors

Els proveïdors són procediments automatitzats que gestionen de forma transparent la pertinença dels usuaris als espais, segons propietats definides per a cada usuari. Per a fer-ho, quan un usuari entra al campus virtual, es comprova si existeix algun proveïdor definit que compleixi amb les propietats d'aquest usuari. Si n'existeixen, s'executen, i aquests s'encarreguen de donar d'alta a l'usuari als espais als que ha de tenir accés.

Per exemple: Proveïdor **AlumneUdL**

Quan entra un usuari al campus virtual, es comprova si té una propietat anomenada "Tipus Usuari". Si aquesta propietat conté el valor "AlumneUdL", s'executa el proveïdor "Alumne UdL", que dona d'alta l'usuari als espais als que han de tenir accés els alumnes de la UdL.

3.1.8 Creació d'espais

Els espais a Sakai tenen associat un tipus, de manera que quan es crea un espai d'un tipus concret, es fa mitjançant una plantilla que inclou unes eines determinades i uns rols amb uns permisos determinats. Els tipus d'espai que trobem a Sakai per defecte són:

- Course sites: Un espai de tipus *course* és l'espai de treball oficial per a un curs particular d'una institució acadèmica, i pot estar enllaçat a una base de dades per tal d'omplir automàticament les seves orles.
- Project sites: Els espais de tipus *projecte* han estat dissenyats per tal de facilitar la col·laboració. En aquests espais, els participants poden convidar qualsevol altre usuari per unir-se a l'espai.

No obstant, de tipus d'espais se'n poden crear més, i les plantilles associades també es poden customitzar.

Actualment, a la UdL utilitzem els tipus següents:

- course: El espai de tipus *course* s'utilitzen per a crear espais d'assignatures. Cadascuna de les assignatures que oferta la UdL té associat un espai de tipus *course* al Campus Virtual.
- projecte docent: Aquest tipus d'espai està pensat per a professors que volen compartir recursos, discussions, missatges, etc. Els professors poden crear-se ells mateixos aquests tipus d'espais i afegir-hi els participants que desitgin.
- projecte: Els espais d'aquest tipus són l'equivalent als anteriors, però per als alumnes. Els alumnes poden crear-se ells mateixos espais de tipus *projecte*.

Evidentment, cadascun d'aquests tipus, té associada una plantilla que indica quines eines per defecte estaran presents a l'espai, quotes permeses, permisos dels usuaris segons els rols amb els que es creen, etc.

En aquest projecte, ens centrarem en els espais de tipus *course*.

Des de l'equip que gestiona el campus, per tal de facilitar l'accés de tota la comunitat a la docència virtual, a l'inici de curs es crea automàticament un espai de tipus curs per cada assignatura planificada a les secretaries de gestió acadèmica.

Aquesta creació automàtica, fins al moment d'aquest projecte es feia mitjançant scripts perl que cridaven serveis web. Inicialment, aquests scripts eren molt senzills, incloent mètodes que, mitjançant consultes a la base de dades de gestió acadèmica, creaven espais d'assignatura, afegien els professors als espais d'assignatura corresponents o bé hi afegien els alumnes a mesura que s'anaven matriculant. Però el Campus Virtual no podia representar estructures de cursos complexes, de manera que aquesta assignació era sempre directament a l'espai d'assignatura, no representàvem cap jerarquia amb grups.

A mesura que va anar passant el temps, els casos d'ús van anar creixent i es van haver d'anar afegint mètodes per a tractar cadascun d'aquests casos, fins al punt que la gestió s'estava complicant molt.

La sincronització amb les dades origen era difícil, ja que qualsevol petit canvi representava sovint un replantejament o ampliació d'aquests serveis web.

D'altra banda, a més de les limitacions tècniques que representava tota aquesta metodologia, cal dir que des del punt de vista del professor també es trobaven

bastants restriccions pel que fa a la gestió dels seus espais, doncs fins al moment el professor no havia pogut decidir mai com organitzar-se els espais dins el Campus Virtual, ni gestionar els seus grups de matrícula com volgués, ni podia gestionar els participants de l'espai, ni tant sols podia crear-se un espai dins el campus.

Amb tot això, i aprofitant el canvi de versió de Sakai, de la 2.7 a la 2.9, es va pensar en una nova manera de sincronitzar les dades del sistema origen de gestió acadèmica amb el Campus Virtual.

Calia doncs aconseguir una connexió més directa i millorada amb el sistema origen que repercutís en una gestió més fàcil i oberta dels espais del Campus Virtual tant per part dels administradors del campus com dels professors.

Ja en la versió 2.4 de Sakai, va aparèixer el concepte '*Course Management*' (gestió de cursos), una API de Sakai que proporciona suport per a la definició d'una estructura de cursos, de manera que podríem integrar en espais de tipus course conceptes d'aquesta estructura, com per exemple els grups de matrícula, i les eines d'aquests tipus d'espais podrien aprofitar aquesta informació.

Capítol 4

Course Management: El model teòric

Fins ara, els espais de tipus curs a Sakai no treuen prou profit de les dades acadèmiques per a poder representar estructures de cursos complexes. Sí que n'obtenim la informació, però no disposem d'una estructura a Sakai per a poder-la representar i usar d'una forma flexible. A més, actualment no hi ha una forma automatitzada per crear espais a partir dels diversos grups d'alumnes d'una assignatura.

Aquesta integració deficient entre el Campus Virtual i els sistemes origen de les dades té un impacte negatiu en tota la comunitat universitària que en fa ús.

Els professors es veuen limitats en l'ús i la gestió dels seus propis espais, mentre que el personal de suport es veu obligat a treballar al voltant de les limitacions del sistema, sovint perdent el temps en fer tasques manuals repetitives.

Tots aquests problemes, repercuteixen en els beneficis que els estudiants n'haurien de treure de l'ús del Campus Virtual.

La solució passa per implementar una millor integració entre el Campus Virtual i els sistemes de dades origen. Si realitzem una definició adequada de les relacions entre l'estructura de cursos i els espais del campus, aconseguirem que

aquests espais siguin manegats més fàcilment tant pels administradors del sistema com pel professorat i això repercutirà en un benefici directe també per als estudiants.

D'altra banda, posarem a disposició del professorat la possibilitat de crear nous espais on agrupar els seus alumnes de la forma més convenient per a la seva docència, en cas que l'espai creat automàticament no li sigui d'utilitat amb l'estructura estandarditzada.

4.1 Objectius

El propòsit del Course Management és definir un model de servei que representi l'organització de cursos en institucions d'ensenyament superior. Podem resumir els objectius concrets del projecte en:

- Crear una única API de Course Management que modeli una organització de cursos sense cap dependència amb altres serveis de Sakai, però podent tenir com a clients altres serveis ja existents (l'eina d'informació de l'espai, l'eina orla, etc.)
- Proporcionar un servei de gestió de cursos fiable, mitjançant el qual les eines de Sakai poden mantenir les dades sincronitzades amb la informació externa de cursos.

No obstant, la recerca d'aquesta estructura jeràrquica que ha de servir per qualsevol model organitzatiu de qualsevol institució d'educació superior, topa amb les següents limitacions:

- Les definicions varien molt entre les diferents institucions, per exemple, la temporalització dels cursos pot ser trimestral, quadrimestral, semestral, ...

- L'estructura de la mateixa institució pot variar entre universitats (Campus, Departaments, Facultats, ...)
- La planificació de cada curs concret pot variar entre diferents universitats (Assignatures, Grups, ...)

L'objectiu principal del Course Management és trobar una estructura genèrica capaç de modelar els diferents escenaris de cada institució.

4.2 Conceptes bàsics

L'estructura de cursos de Sakai es basa en un sistema de contenció.

Un CourseSet descriu una col·lecció de CanonicalCourses. Els CanonicalCourses poden contenir d'altres CanonicalCourses o bé CourseOfferings. Els CourseOfferings contenen Sections. Les Sections poden contenir d'altres Sections (sub-sections).

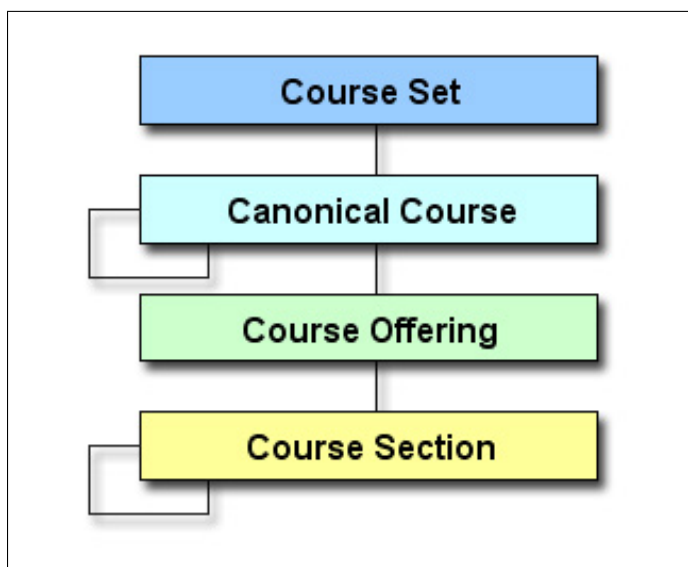


Figura 4.1: Relació jeràrquica entre els objectes del course management

Les relacions entre els diferents objectes que conformen el course management es mostra a continuació:

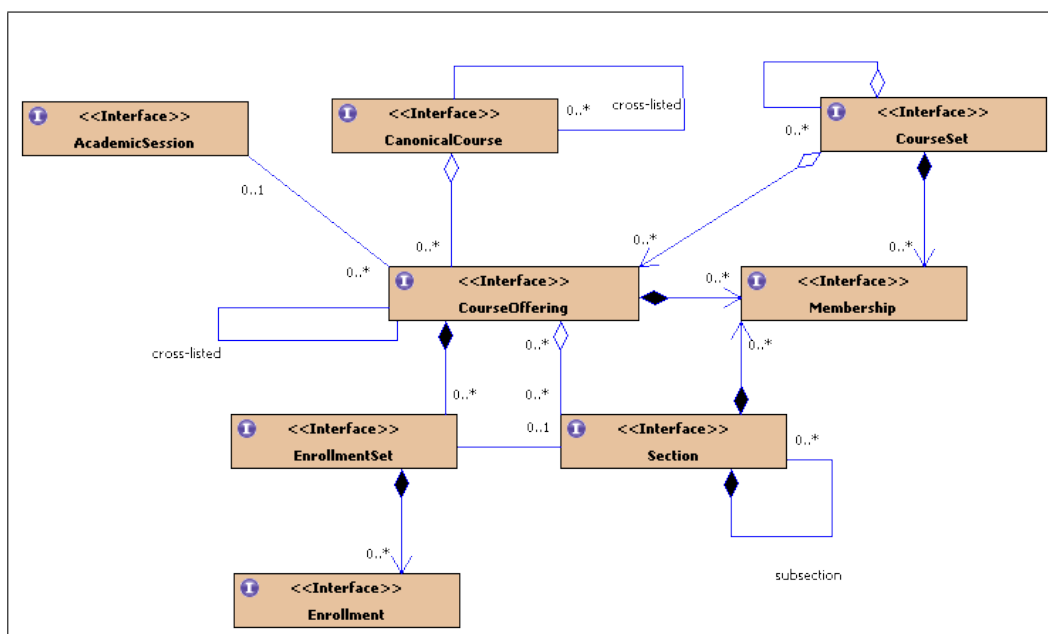


Figura 4.2: Diagrama de classes del Course Management

4.2.1 CourseSet

És una col·lecció de Canonical Courses i/o Course Offerings que es pot utilitzar per descriure departaments, titulacions, centres, ... i d'altres tipus d'agrupacions de cursos d'alt nivell.

Proporcionen una forma d'agrupar els CanonicalCourses en col·leccions útils. De tota manera no hi ha cap requeriment pel qual un CanonicalCourse hagi de pertànyer a un CourseSet.

Un CourseSet pot contenir d'altres CourseSets per tal de suportar una jerarquia del tipus Facultat - Departament - Àrea de coneixement.

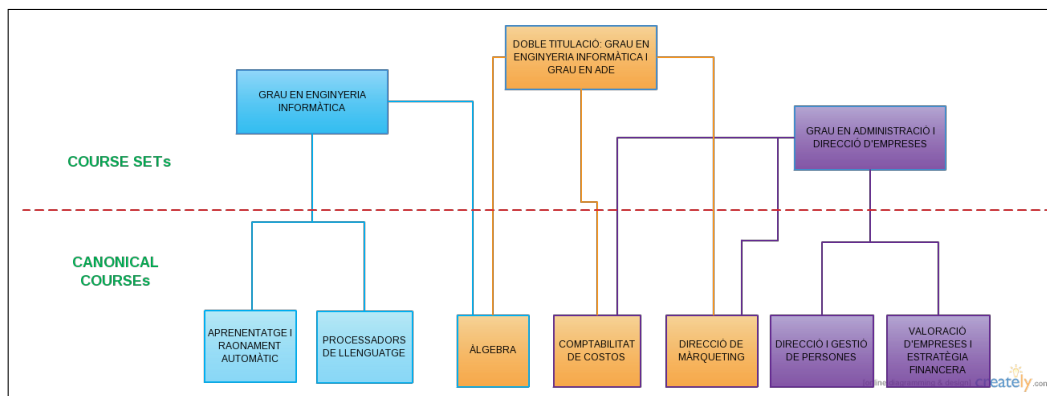


Figura 4.3: Exemple de la relació jeràrquica entre CourseSets i CanonicalCourses

4.2.2 CanonicalCourses

Els Canonical Courses són la definició abstracta d'una instància particular d'un curs. Es poden pensar com cursos, assignatures o matèries que s'ofereixen a la universitat, l'estructura d'assignatures en general, sense tenir en compte si s'ofereixen en un determinat quadrimestre o no.

S'utilitzen per definir les propietats d'una assignatura que no varien al llarg dels quadrimestres i anys acadèmics (nom, crèdits, ...)

4.2.3 CourseOffering

És un curs que s'ofereix en un determinat quadrimestre / semestre. Per evitar confusions, es defineix també un objecte AcademicSession que descriu un espai de temps dins un any acadèmic.

Per tant, un Course Offering és un Canonical Course associat a una determinada Academic Session. Pot tenir un conjunt de membres associats (professors, teaching assistants, ...). Cada membre associat a un CourseOffering té un rol.

Els estudiants matriculats a un course offering, estaran assignats a una o varies Sections derivades del curs, i estaran associats a un o varis EnrollmentSets agre-

gats a aquestes Sections.

La llista d'estudiants associada al course offering és una agregació de tots els estudiants matriculats en cadascuna de les sections derivades d'aquest curs.

4.2.4 Section

Les Sections són probablement l'objecte més important definit en el Course Management.

Una Section és la manera de representar un grup de persones que es troben en un determinat espai de temps al mateix lloc. Pot contenir totes les persones que participen en la classe / curs, o bé pot contenir subgrups.

Les Sections poden ser de diferents tipus, per tant, una persona pot pertànyer a més d'una section associada a un determinat course offering.

Les Sections es poden associar també amb un conjunt de Meetings que descriuen on i quan es troben els membres de la Section. També poden definir una mida màxima que indica el nombre màxim d'estudiants permesos.

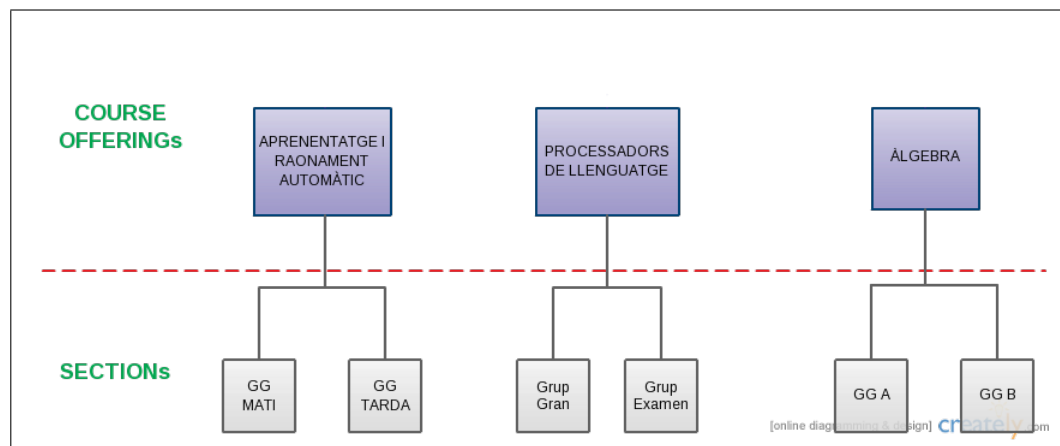


Figura 4.4: Exemple de la relació jeràrquica entre CourseOfferings i Sections

4.2.5 EnrollmentSet

Els EnrollmentSets modelen col·leccions d'Enrollments per una determinada Section o CourseOffering, i emmagatzemen també d'altres relacionades amb la matrícula com són els OfficialInstructors (professors) i DefaultEnrollmentCredits (crèdits que s'obtindran en superar l'assignatura).

Aquesta informació es separa de les Sections (grups), ja que algunes institucions poden tenir Sections sense crèdits associats i d'altres institucions, per exemple, poden tenir CourseOfferings sense Sections.

4.2.6 Enrollment

Un Enrollment representa la relació oficial entre un estudiant i "alguna cosa que té una qualificació final", com per exemple, una matrícula.

Conté informació sobre si un estudiant s'ha donat de baixa, crèdits, estat, etc.

4.2.7 Membership

Els Membership relacionen un usuari amb un rol particular dins un curs. Per exemple, un usuari pot tenir el rol de:

- Administrador en un CourseSet
- Teaching Assistant en una Section
- Estudiant en una altra Section
- Professor en un CourseOffering

Els Membership tenen la propietat Status, que pot ser usada per indicar estats de persones que poden definir les mateixes institucions.

4.2.8 Academic Session

Representa una temporalització (quadrimestres, semestres, anys,...). Té la propietat “current” que indica si es tracta de l’espai de temps actiu en aquest moment.

Tot i que la majoria d’institucions només tenen actiu una AcademicSession alhora, el servei proporciona un mètode per tal que ens retorni una llista de AcademicSessions. Això ens permet gestionar casos d’ús com els següents:

- Algunes universitats ofereixen cursos d’estiu de durada variable.
- Institucions molt grans poden comprendre varies facultats / escoles o mantenir diferents programes de graduació amb diferents calendaris.
- Els entorns d’aprenentatge online poden haver de suportar sessions escalonades, o sessions autodidactes en les quals l’inici i el fi són determinats per objectius.

4.3 Casos d’ús

En la recerca i anàlisi del Course Management es va demanar a diferents universitats que definissin els casos d’ús que tractaven, referents a l’organització jeràrquica dels objectes definits anteriorment, i la traducció que en feien cap a espais de Sakai. Els resultats van ser els següents:

- **Patró simple:** Una assignatura s’associa amb un únic espai, els alumnes es matriculen al curs i no hi ha grups.

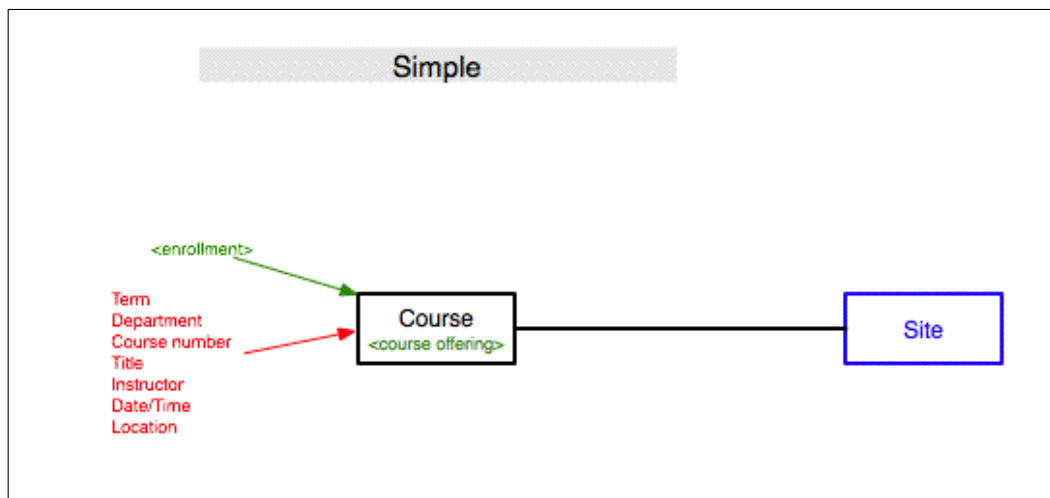


Figura 4.5: Cas d'ús: Patró simple

- **Patró amb Sections i Enrollments al CourseOffering:** Una única assignatura amb varis grups associada amb un únic espai amb estudiants i professors encabits dins les diferents Sections. Els enrollments s'associen en aquest cas directament al CourseOffering. Les eines de l'espai poden publicar o limitar per Sections.

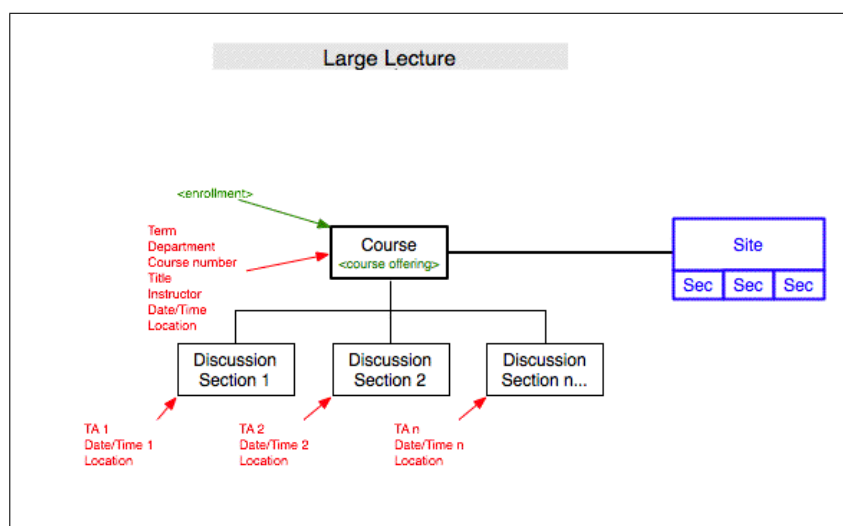


Figura 4.6: Cas d'ús: Patró amb Sections i Enrollments al CourseOffering

- **Patró amb Sections i Enrollments a les mateixes Sections:** Una assignatura amb diferents grups s'associa a un únic espai amb les sections definides en el sistema origen. Els estudiants es matriculen directament al grup concret, per la qual cosa els enrollments s'associen en aquest cas a les sections. Les eines de l'espai poden publicar o limitar per sections.

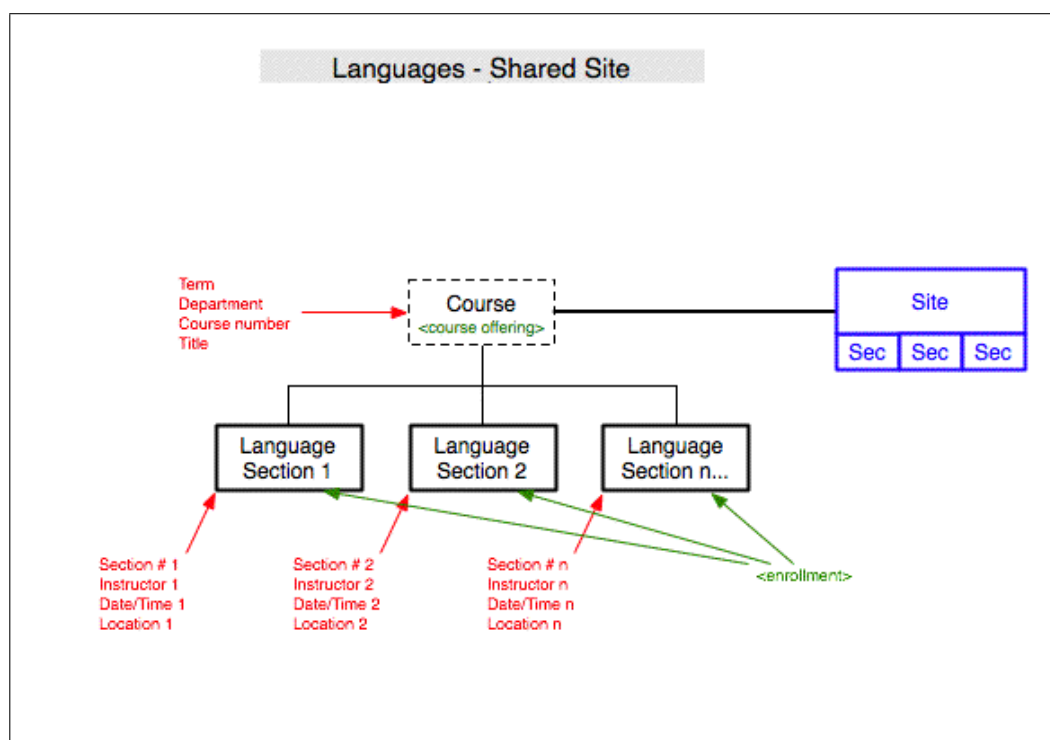


Figura 4.7: Cas d'ús: Patró amb Sections i Enrollments a les mateixes Sections

4.4 Integració amb Course Management

El Course Management és principalment una API. Una descripció d'estructures de cursos oficial.

Per tant, es pot pensar en el Course Management com un contracte d'integració i sincronització entre Sakai i l'empresa. Aquesta API conté dos interfícies:

- ***CourseManagementService*** de només lectura, i és la única usada per altres components i aplicacions de Sakai.
- ***CourseManagementAdministration*** és opcional, si s'implementa, aporta una manera d'actualitzar les dades emmagatzemades en el mateix course management.

4.4.1 Clients de la API CM

L'API del CM s'empra en les següents àrees de Sakai:

- Integració amb Authorization, Groups i Section Info tool: Quan una assignatura s'ha associat amb un espai de tipus course, la informació de les sections (grups) i memberships (participants) es pot afegir automàticament a l'espai. Els enrollments del courses i sections (juntament amb els memberships que no són estudiants) es veuran reflexats automàticament en l'espai. L'instructor pot decidir mantenir aquesta sincronització dels grups i participants amb el course management o desconnectar aquest espai del course management, de manera que ja no s'actualitzaran de les dades. Això ho pot fer a l'eina Section Info tool canviant de auto-managed a self-managed sections, sempre i quan els administradors permetin aquesta acció mitjançant la configuració.

- Integració amb l'eina Orla: La informació sobre els enrollments dels estudiants (estat, crèdits, etc) es poden mostrar en l'eina Orla. Aquesta eina ens permet llistar participants del grup seleccionat o bé de tot l'espai.

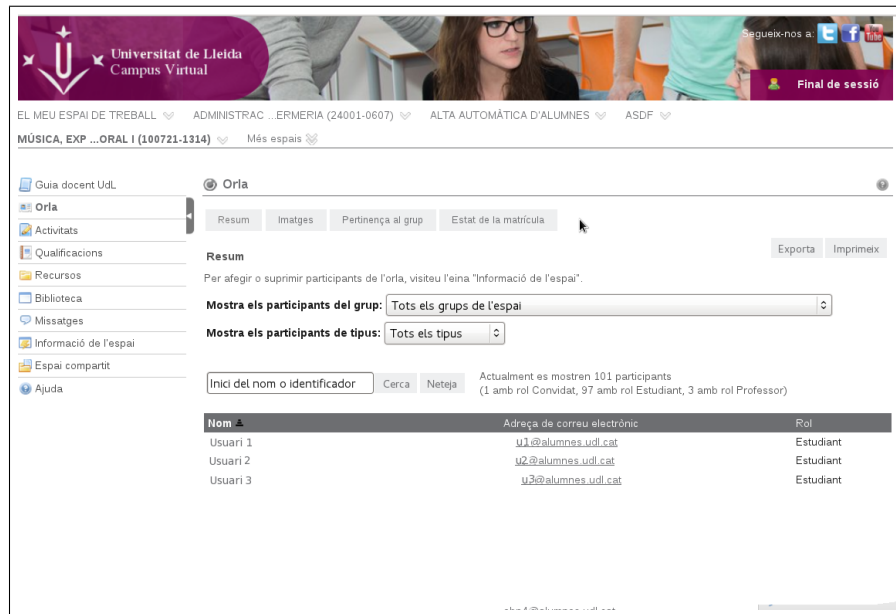


Figura 4.8: Exemple d'eina orla

- Integració amb l'eina Informació de l'espai: Aquesta eina ens permet, si tenim un rol amb permisos suficients, veure tots els participants de l'espai, els crèdits que cada participant ha matriculat i el grup o grups als que pertanyen. Si tenim prou permisos, aquesta eina ens permet configurar l'espai, afegint o traient eines, afegint o traient grups, etc.

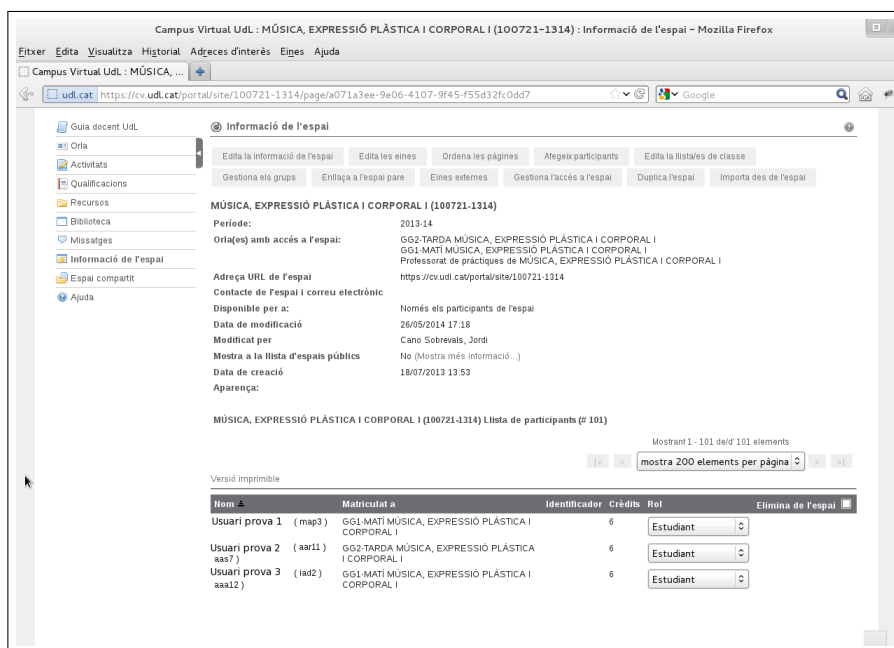


Figura 4.9: Exemple d'eina Informació de l'espai

- Integració amb l'eina Configuració de l'espai: Els professors poden crear sites com a combinacions dels cursos i sections que tinguin assignats. Es restringeix la creació de sites a cursos i sections associats a una Academic-Session que sigui “current”.



Figura 4.10: Pantalla inicial en la creació d'un nou espai mitjançant l'eina Configuració de l'espai.

Si crea un espai de tipus curs, el següent pas mostra tots els grups on el professor està planificat i, per tant, en pot disposar per a configurar el nou espai.

Configuració d'espais

Informació sobre el curs/la secció
 Selecció de Cursos/Seccions - 2013-14
 Seleccioneu el curs/la secció apropiat per afegir al vostre espai. Si seleccioneu classes/seccions múltiples es combinaran en un sol espai.

MÚSICA, EXPRESSIÓ PLÀSTICA I CORPORAL I (100721)

- ☐ GG1-MATÍ MÚSICA, EXPRESSIÓ PLÀSTICA I CORPORAL I (76742) Grup de matrícula (Un espai que inclou aquest codi de curs ja existeix, seleccionar)
- ☐ GG2-TARDA MÚSICA, EXPRESSIÓ PLÀSTICA I CORPORAL I (76743) Grup de matrícula (Un espai que inclou aquest codi de curs ja existeix, seleccionar)
- ☐ Professorat de pràctiques de MÚSICA, EXPRESSIÓ PLÀSTICA I CORPORAL I (TA-100721) Grup de professors ajudants (Un espai que inclou aquest codi)
- ☐ Alumnus de Grau en Educació Infantil-100721 (TIT-G401-100721) Grup de titulació

MÚSICA, EXPRESSIÓ PLÀSTICA I CORPORAL III (100723)

- ☐ GG1-MATÍ MÚSICA, EXPRESSIÓ PLÀSTICA I CORPORAL III (81728) Grup de matrícula (Un espai que inclou aquest codi de curs ja existeix, seleccionar)
- ☐ GG2-TARDA MÚSICA, EXPRESSIÓ PLÀSTICA I CORPORAL III (81729) Grup de matrícula (Un espai que inclou aquest codi de curs ja existeix, seleccionar)
- ☐ Professorat de pràctiques de MÚSICA, EXPRESSIÓ PLÀSTICA I CORPORAL III (TA-100723) Grup de professors ajudants (Un espai que inclou aquest codi)
- ☐ Alumnus de Grau en Educació Infantil-100723 (TIT-G401-100723) Grup de titulació

DIDÀCTICA DE L'EDUCACIÓ ARTÍSTICA (100819)

- ☐ GG3-TARDA DIDÀCTICA DE L'EDUCACIÓ ARTÍSTICA (81601) Grup de matrícula (Un espai que inclou aquest codi de curs ja existeix, seleccionar igualment)
- ☐ Alumnus de Grau en Educació Primària-100819 (TIT-G402-100819) Grup de titulació

Figura 4.11: Llistat de sections planificades per a un professor que poden ser afegides a l'espai creat.

Capítol 5

Course Management: La implementació a la UdL

En aquest capítol s'explica tot el procés d'anàlisi, disseny i implementació que s'ha dut a terme en la integració del Course Management al Campus Virtual de la UdL.

En la fase d'anàlisi, s'ha realitzat un estudi de tots els tipus d'espais que hi ha al Campus Virtual en el curs acadèmic 2012-13 , per tal de poder decidir quins són aquells tipus d'espais que per les seves característiques són susceptibles de ser integrats amb el Course Management.

Seguidament, un cop estudiat el model teòric del Course Management i els casos d'ús que apliquen a la UdL, s'ha dut a terme la fase de disseny on s'han pres decisions que afecten la implementació del Course Management i la sincronització de les dades amb els sistemes origen.

Finalment, en la implementació es descriu les tecnologies emprades i els processos implementats.

5.1 Anàlisi

Durant aquest estudi que hem comentat s'ha definit les següents característiques per cadascun dels tipus d'espais que hi ha creats al Campus:

- **Descripció breu:** Breu explicació sobre la finalitat de l'espai.
- **Rols participants:** Rols definits al Campus Virtual que hi participen.
- **Usuaris participants:** Tipus d'usuari que hi pot accedir.
- **Divisió en grups:** Si l'espai permet o ha de permetre la divisió en grups.
- **Responsable de la creació:** Qui n'és el responsable de la creació i amb quina freqüència cal fer-ho.
- **Cicle de vida:** Quant de temps cal mantenir-lo accessible al Campus Virtual.
- **Proveïdor de les dades:** Indica si tenim identificat el sistema origen de les dades.

Del resultat d'aquest estudi de tipologies d'espais n'hem extret la següent informació:

Espais tipus course

Descripció	Espais destinats a donar suport a la docència i l'aprenentatge d'assignatures reglades de la UdL.
Rols	Instructor, Teaching Assistan, Student, Convidat
Usuaris	Alumnes UdL, Professors UdL, Professors Externs i Usuaris Genèrics
Divisió en grups	Sí, des del moment de la creació
Responsable creació	ASIC ,procés automàtic
Cicle de vida	Actiu durant l'any acadèmic actual i inactiu els dos anys acadèmics següents
Proveïdor identificat	Sí, UXXI

Espais tipus titulació

Descripció	Espais destinats a donar suport als membres d'una titulació i a compartir informació per diferents canals
Rols	Coordinador, professor i estudiant
Usuaris	Alumnes UdL, Professors UdL i Professors externs
Divisió en grups	Sí, cada espai de titulació té un grup per cadascuna de les assignatures planificades. Els membres de cada grup són els professors i alumnes.
Responsable creació	ASIC, procés automàtic
Cicle de vida	Actiu durant l'any acadèmic actual i inactiu els dos anys acadèmics següents
Sistema origen identificat i acotat	Sí, UXXI

Espais tipus centre

Descripció	Espais destinats a la comunicació i compartició de recursos entre membres d'un mateix centre.
Rols	Instructor, Student
Usuaris	Alumnes i Director/a, Degà/na centre.
Divisió en grups	No
Reponsable creació	ASIC, un sol cop a la vida
Cicle de vida	Sempre actius.
Proveïdor identificat	Sí, UXXI

Espais tipus tutoria (Nèstor)

Descripció	Espais destinats a donar suport a sistema de tutories entre professors i alumnes.
Rols	Instructor, Teaching Assistant, Student
Usuaris	Alumnes de primer any de la UdL i Professors UdL
Divisió en grups	sí, Cada professor té un grup amb el seu nom on es recullen els alumnes que tutoritza. Hi ha tans grups com tutors.
Reponsable creació	ASIC, procés automàtic
Cicle de vida	Actiu durant l'any acadèmic actual, més el que tardin a acabar la titulació. Aquest espai serveix per a tutoritzar l'alumnat durant tota la seva carrera
Sistema origen identificat i acotat	No, depenem d'uns fitxers que es defineixen manualment a les secretaries

Espais tipus comunitats

Descripció	Espais destinats a la comunicació i compartició de recursos entre membres amb interessos comuns.
Rols	maintain i access
Usuaris	Qualsevol usuari del campus
Divisió en grups	No
Reponsable creació	ASIC, un sol cop a la vida
Cicle de vida	Sempre actius.
Proveïdor identificat	No

Espais tipus projecte

Descripció	Espais destinats a compartir informació i donar mitjans per millorar la comunicació entre membres d'un projecte.
Rols	maintain i access
Usuaris	Qualsevol usuari del campus
Divisió en grups	No
Reponsable creació	ASIC, procés manual
Cicle de vida	Sempre actius.
Proveïdor identificat	No

Espais tipus intranet

Descripció	Espais destinats a emmagatzemar informació pública pels membres de la UdL.
Rols	maintain i access
Usuaris	Alumnes, PDI i PAS
Divisió en grups	No
Reponsable creació	ASIC, procés manual
Cicle de vida	Sempre actius.
Proveïdor identificat	Sí, LDAP, però només ens permet distingir entre Alumnes o PDIPAS

Un cop acabat aquest estudi ens adonem que hi ha 3 tipus d'espais susceptibles de ser integrats amb Course Management: espais de tipus course, espais de titulació i espais de tipus centre. Ja que aquest tipus d'espais utilitzen grups, defineixen una estructura jeràrquica, segueixen una metodologia iterativa ben definida pel que fa a les altes i baixes, i a més, tenim el sistema origen ben identificat i acotat.

Els espais de tipus centre però, tenen unes estadístiques d'ús molt baixes o fins i tot nul·les en molts dels casos, per tant es decideix descartar la integració per aquest tipus d'espai.

5.2 Disseny

En el moment de realitzar aquest projecte, s'està preparant una nova versió de Sakai que serà desplegada al Campus Virtual de la UdL quan canviem d'any acadèmic. Als voltants del mes de Juliol és quan s'acostuma a fer el canvi d'any acadèmic al Campus Virtual, procés que implica la creació dels nous espais d'assignatura per l'any acadèmic següent. Però en aquest cas, prèviament a la creació dels nous espais es farà la migració a la versió 2.9.2 de Sakai.

De l'estudi previ sobre els tipus d'espais que s'integraran amb el Course Management explicat en l'apartat 5.1 se'n deriva que la integració es faci sobre els espais de docència reglada i sobre els espais de titulació.

5.2.1 Matching de conceptes

A priori, estudiat d'una banda l'estructura de cursos que proveeix el Course Management i de l'altra la que n'obtenim del sistema origen, decidim el següent mapping entre les entitats del Course Management i les de Gestió Acadèmica (pel que fa a l'estructura jeràrquica de cursos):

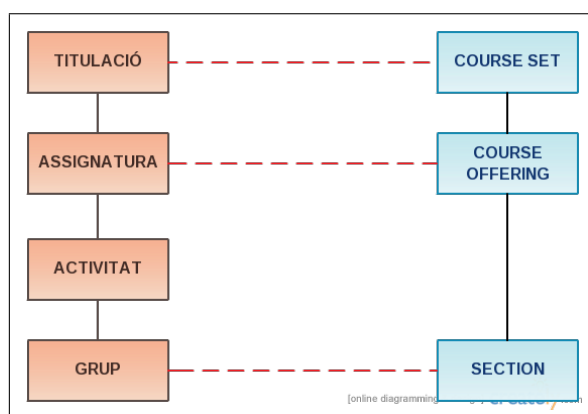


Figura 5.1: Relació jeràrquica de conceptes

A més, com que els CourseSets poden tenir parents, aquest mapping ens permet també afegir l'organització de les titulacions per departaments o per centres si cal.

Decidim no fer ús dels CanonicalCourses, ja que en cas de la UdL no aplica. La informació necessària la podem encabir en els CourseOffering i a més, no podem garantir que la informació que posaríem als CanonicalCourses no variï en el temps.

Per tal de comprovar si el matching de conceptes decidit ens servirà, estudiem els diferents casos d'ús que podem identificar pel que fa a la planificació d'una assignatura:

5.2.1.1 CAS D'ÚS 1: Assignatura ofertada amb un únic grup de matrícula

Es tracta del cas més senzill, una assignatura que s'oferta amb únic grup de matrícula. Traduirem aquest cas d'ús a una única Section (grup de matrícula) dins del CourseOffering (assignatura) ofertat en el CourseSet (titulació).

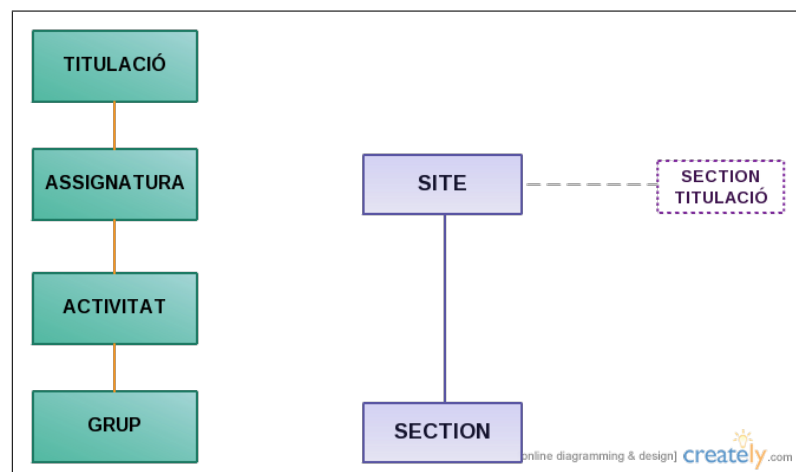


Figura 5.2: Cas d'ús 1 UdL

Deixarem preparat per aquest cas un únic grup de titulació.

La Section tindrà un EnrollmentSet associat que contindrà tots els estudiants matriculats (Enrollments) i els professors oficials de l'assignatura.

Cadascun dels membres que formen part de l'assignatura s'afegiran al nivell corresponent en la jerarquia amb el rol pertinent.

5.2.1.2 CAS D'ÚS 2: Assignatura compartida

Es tracta de dos assignatures ofertades en dos titulacions diferents que comparteixen el grup de matrícula, ja que en realitat els alumnes van junts a classe.

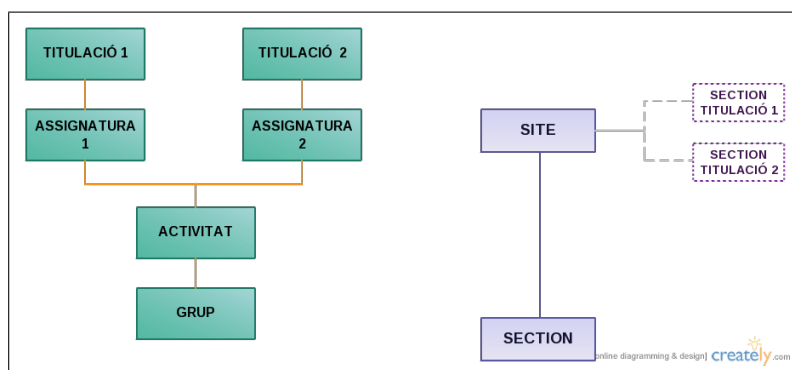


Figura 5.3: Cas d'ús 2 UdL

Aquest cas el traduirem al CourseManagement com un únic grup de matrícula (Section), però deixarem preparats per a un possible ús per part dels professors dos grups d'alumnes segons la titulació que cursen.

5.2.1.3 CAS D'ÚS 3: Assignatura compartida amb més d'un grup de matrícula

Representa dos assignatures ofertades en dos titulacions diferents que a més, tenen planificats dos grups de matrícula.

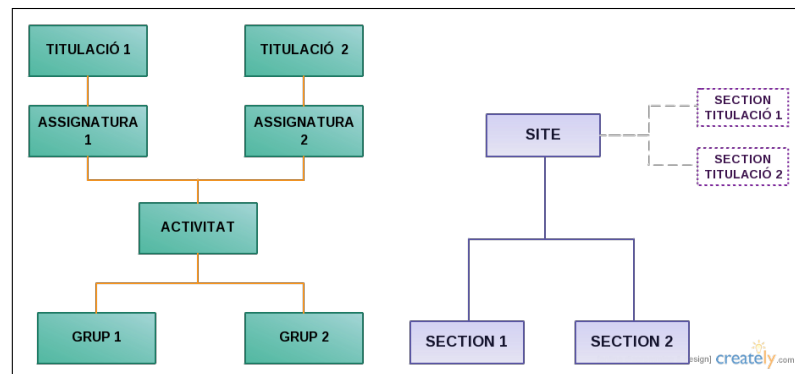


Figura 5.4: Cas d'ús 3 UdL

Aquest cas el traduïm al Course Management amb dos Sections representant els dos grups de matrícula que estaran presents en l'espai d'assignatura creat i d'altra banda, dos grups de titulació que deixem preparats per a un possible ús.

5.2.1.4 CAS D'ÚS 4: La mateixa assignatura ofertada en dos titulacions diferents

Idealment aquest cas no hauria de passar, doncs cada titulació hauria de tenir la seva pròpia codificació d'assignatures, no obstant hem identificat també aquest cas d'ús.

Aquest cas el traduïm al Course Management amb una Section representant l'únic grup de matrícula que té l'assignatura (CourseOffering) i dos grups de titulació que deixem preparats per a un possible ús.

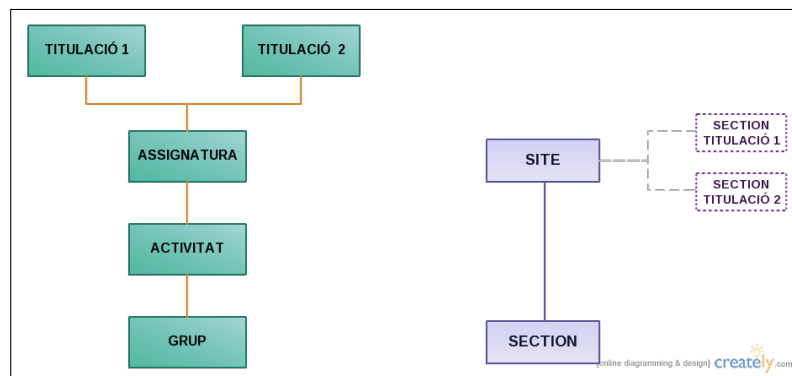


Figura 5.5: Cas d'ús 4 UdL

5.2.2 Consideracions per la implementació a la UdL

Abans de començar a implementar, s'ha pres un seguit de decisions sobre com abordarem aquesta integració.

1. Donat el volum de **dades** que cal tenir connectades en tot moment i l'entorn crític que suposa el sistema origen en certs moments de l'any, es decideix permetre un decalatge d'un dia en la sincronització de les dades del Campus Virtual i aquest sistema origen, de manera que a la nit, es farà una còpia de totes les dades.

Pel que fa al disseny del model de dades del Course Management, decidim optar per customitzar els fitxers hibernate per tal que apuntin al nostre propi model del Course Management.

S'opta per hibernate perquè es tracta d'una tecnologia ja integrada amb el Campus Virtual que permet mapejar els objectes de la base de dades directament amb els objectes del Course Management.

2. Es decideix **reescriure la implementació de l'API** del Course Management per adaptar-la al nostre propi model de dades.

50CAPÍTOL 5. COURSE MANAGEMENT: LA IMPLEMENTACIÓ A LA UDL

3. Pel que fa a l'eina **Sections**, es decideix no fer-ne ús, ja que el propi funcionament d'aquesta eina no permet tenir un sections connectades a un sistema origen i sections no connectades alhora, amb la qual cosa perdríem el sentit del Course Management.

A més, existeix l'alternativa de l'eina Grups, i els professors que vulguin crear-se altres grups, ho podran fer mitjançant aquesta eina disponible a la Informació de l'espai.

4. Pel que fa a l'eina **SiteManage**, es permet que els professors puguin crear-se d'altres espais tipus course, combinant d'altres cursos i sections que ells mateixos tinguin assignats.
5. S'opta per **no implementar els Canonical Courses**, doncs la informació necessària la trobem ja en les altres entitats i no podem garantir que la informació no variï en el temps.
6. Estudiats el sistema origen i els diferents casos d'ús, es creu oportú **no representar totes les activitats planificades**, ja que els grups que se'n deriven no aporten cap informació útil, doncs els alumnes només es matriculen als grups teòrics i per tant, no té sentit poblar els espais del Campus amb Sections buides que a més no representen la realitat de les aules, ja que només es tracta d'aproximacions.
7. En el cas de la UdL no es planifiquen assignatures sense grups, els alumnes sempre s'han de matricular en un grup concret dels que hi ha planificats, per tant, podrem afegir membres amb el rol pertinent a qualsevol nivell de la jerarquia (Membership) però **només tindrem Enrollments a les Sections**.

8. Es definiran els següents **rols per tipus d'espais**:

Docència reglada:

- Professor: És el rol amb més permisos dins un espai d'assignatura. Si s'associa un participant amb rol professor directament a un CourseSet, aquest participant tindrà accés a totes les Sections de tots els CourseOffering que penguin d'aquest CourseSet, i anàlogament passa si l'associem a un CourseOffering. En canvi, si l'associem a una Section, només tindrà accés a aquesta Section.
- Professor ajudant: És un professor que només se li ha assignat grups de pràctiques.
- Estudiant: Només podem associar participants amb rol Estudiant a les Sections, i evidentment, els permisos que li atorguen aquest rol són més limitats que els de Professor o Professor ajudant.
- Convidat: Es fa servir per als usuaris genèrics.

Espais de titulació:

- Coordinador: És el rol que se li atorga al coordinador d'una titulació, s'assigna directament al CourseSet que representa la titulació.
- Docent: Representa un professor en una assignatura de la titulació.
- Estudiant: Estudiant de qualsevol assignatura que ofereix la titulació.
- Convidat: S'utilitza per als usuaris genèrics.

9. S'afegirà també una Section on s'hi inclourà els **professors de pràctiques**, de manera que es podran crear els seus grups de pràctiques.

10. Com en anys anteriors, caldrà afegir als espais **usuaris genèrics** per tal de poder accedir a les assignatures mentre els alumnes encara no han formalitzat la matrícula. Aquesta creació es farà mitjançant un job de Quartz¹.
11. En la planificació d'una assignatura es permet programar un **professor extern** (no UdL) en qualsevol dels seus grups. Aquests usuaris externs no tenen cap compte que pugui validar-los a la UdL, per tant, cal crear-los-hi un compte al Campus Virtual. Es programarà un altre job de Quartz que quan detecti la planificació d'un professor extern, crei un usuari al Campus Virtual.
12. Si es planifiquen **grups nous** un cop els espais ja s'han creat, notificarem el professor afectat per tal que decideixi si vol afegir el grup al seu espai, doncs a partir d'ara ells en són els gestors.
13. S'aprofita aquesta implementació per desenvolupar un concepte que s'ha demanat repetidament per part del professorat. En certes ocasions el professorat ha manifestat que els resultaria útil poder tenir agrupats els alumnes per la titulació que cursen enlloc de tenir-los agrupats pel grup de matrícula al que s'han inscrit.

Decidim d'una banda crear els espais d'assignatura amb la informació literal que hi ha al sistema origen, és a dir, els grups de matrícula tal i com estan planificats. No obstant, es deixa preparat per a un possible ús els **grups d'alumnes agrupats per titulació**, de manera que si un professor vol configurar-se el seu espai de treball amb aquest tipus de grups, ho pot fer des de l'eina Informació de l'espai.

¹Quartz és un framework de scheduling open source que proporciona funcionalitats avançades per la calendarització de tasques en Java.

14. Es crearà automàticament els espais de les assignatures ofertades amb les Sections corresponents als grups de matrícula.

La **creació automàtica d'espais de tipus course** es farà mitjançant un job de Quartz que cal programar. Programarem un job per la creació automàtica dels espais de tipus assignatura i un altre per la creació automàtica dels espais tipus titulació, ja que les plantilles associades per a crear-los són diferents.

15. La informació sobre participants, rols i sections a les que pertanyen es desa als realms dels espais. Aquests realms s'actualitzen quan es desen, per aquest motiu, si hi ha canvis respecte els participants d'una section o dels rols que tenen assignats, no es veuran reflectits fins que no es desi el realm.

Cal doncs **actualitzar els realms dels espais** que estan integrats amb el Course Management. La implementació del Course Management inclou un job de Quartz que fa aquesta actualització, no obstant decidim modificar-lo per tal que actualitzi només els realms dels espais que han sofert alguna modificació agilint-ne així l'execució.

5.3 Implementació

En aquest apartat s'explica com s'ha dut a terme la implementació del Course Management en el cas concret de la UdL, tenint en compte el model teòric explicat i les consideracions preses per a la implementació.

Es separa en l'explicació la implementació de l'API del Course Management i el procés que s'ha seguit per a la sincronització de les dades origen amb el model de la UdL dissenyat.

5.3.1 Implementació de l'API del Course Management

Basant-nos en la implementació de referència del Course Management, com ja s'ha explicat anteriorment, hem fet els canvis pertinents per adaptar-ne el model a les nostres necessitats.

El mapa de relacions entre classes del Course Management queda de la següent manera:

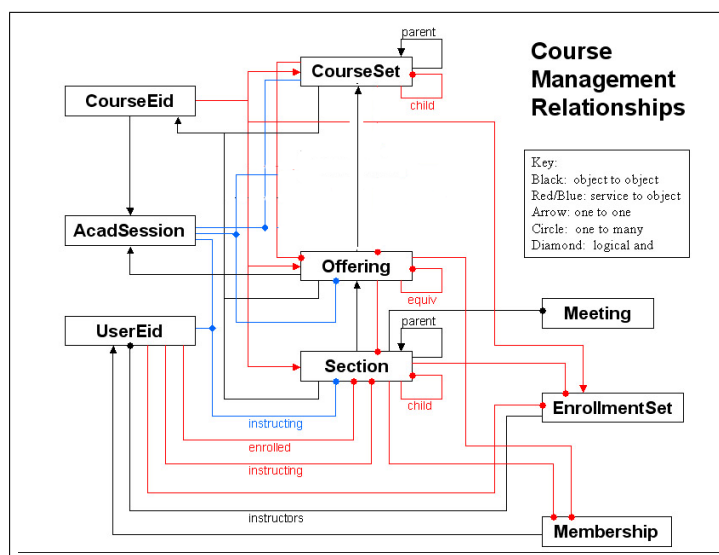


Figura 5.6: Entitat relació resultant de les classes del Course Management en la implementació a la UdL

La implementació de les diferents classes i el servei del Course Management s'ha realitzat en Java, ja que l'objectiu és integrar-ho al Campus Virtual de la UdL, el qual es basa principalment en Java, tot i que engloba d'altres tecnologies com Spring, Hibernate, JSF, etc.

S'ha optat per fer la persistència d'aquestes classes amb Hibernate ja que es tracta d'una tecnologia ja integrada a Sakai.

Hibernate és una eina que facilita el mapeig d'atributs entre una base de dades relacional i el model d'objectes d'una aplicació mitjançant la definició en arxius XML.

A continuació es mostra l'esquema resultant de la base de dades:

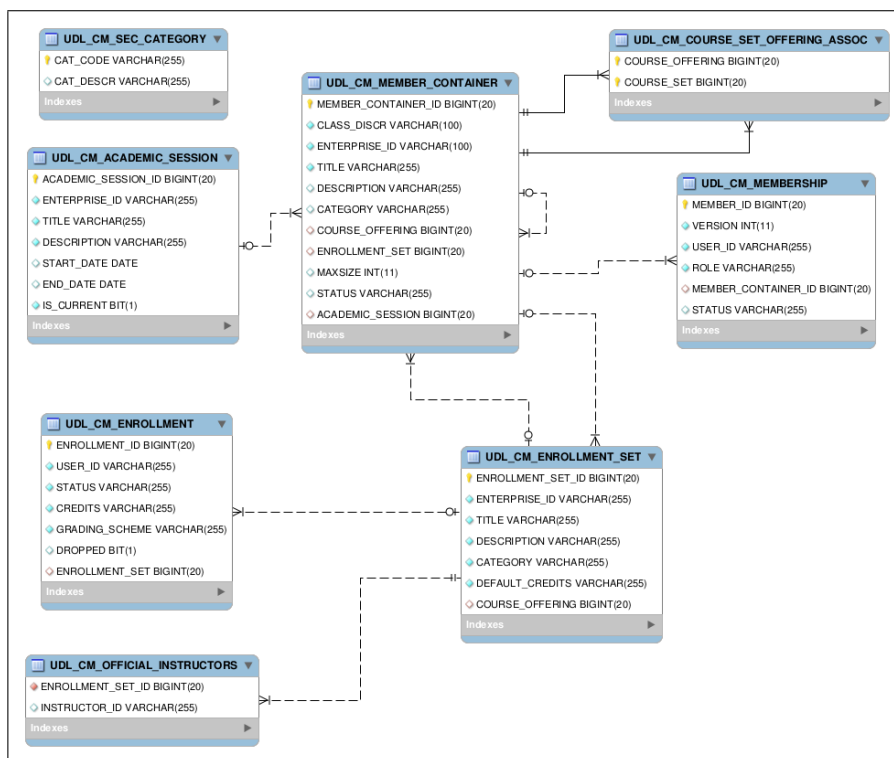


Figura 5.7: Model de dades que utilitzarem per la integració del Course Management a la UdL

Course Management Object - Database Map

Segons el model entitat relació i el model de base de dades, podem establir el següent:

Course Management Object	Database Object
AcademicSession	UDL_CM_ACADEMIC_SESSION
CourseSet	UDL_CM_MEMBER_CONTAINER
CourseOffering	UDL_CM_MEMBER_CONTAINER
CourseOffering/CourseSet Mapping	UDL_COURSE_SET_OFFERING_ASSOC
Section	UDL_CM_MEMBER_CONTAINER
EnrollmentSet	UDL_CM_ENROLLMENT_SET
Instructor List	UDL_CM_OFFICIAL_INSTRUCTORS
Enrollment	UDL_CM_ENROLLMENT
SectionCategory	UDL_CM_SEC_CATEGORY

En l'annex A s'especifica en detall els fitxers inclosos en cadascun d'aquests directoris, i el cd adjunt a aquest projecte inclou tots els fitxers.

5.3.2 Implementació del procés de sincronització

El procés de sincronització dels sistemes origen amb el Course Management s'encarrega de recollir totes les dades necessàries, tractar-les i bolcar-les a les taules de l'esquema que hem creat, omplint així el nostre model.

Gràcies a aquesta informació, professors, alumnes i d'altres usuaris del Campus Virtual poden accedir als espais en els quals estan matriculats i/o planificats.

A continuació s'exposa l'anàlisi, disseny i implementació que s'ha seguit en la construcció d'aquest procés.

5.3.2.1 Anàlisi i disseny del procés de sincronització

Ja des de l'inici es va descartar la sincronització directa al sistema origen, degut al gran volum de dades a tractar i també a que es tracta d'un entorn crític en algunes èpoques de l'any com és, per exemple, la matrícula.

Amb aquesta premisa, es van dissenyar les següents fases per aquest procés:

Fase de còpia: Per a transferir totes les dades de gestió acadèmica a unes taules intermitges ubicades ja a la base de dades del Campus Virtual.

Fase de transformació: Per a fer el tractament d'aquestes dades i encabir-les al model definit del course management.

Aquesta definició tenia però els següents inconvenients:

1. Durant els aproximadament 5 minuts que tardava la fase de transformació, el Course Management es quedava desproveït, de manera que qualsevol imprevist podria fer que el model de dades quedés buit.
2. Si la fase de transformació fallava en qualsevol dels passos que la integren, tot el Course Management es quedava desproveït, i per tant, els espais no tenien informació.
3. Segons aquesta definició, com que el procediment esborrava totes les dades del Course Management i les tornava a carregar i tractar, no podíem saber quins espais/grups havien canviat, per tant, calia actualitzar tots els realms encara que no hi haguessin sofert canvis.

De manera que tot i tractar-se d'un procediment ben senzill, resultava inacceptable que els espais proveïts pel Course Management poguessin quedar sense

grups ni participants en qualsevol moment, i per tant es va optar per redefinir el procés per tal d'afegir-hi seguretat en totes les fases.

La idea del nou procediment és tractar només els canvis (altes, baixes o modificacions) de contenidors (courseSets, courseOfferings o sections) i usuaris (professors o alumnes). S'ha dissenyat aquest nou procediment en cinc fases, i a més s'ha modificat el job de Quartz que actualitza els realms per tal que ho faci només d'aquells espais o grups en els que hi ha hagut canvis d'usuaris al sistema origen.

Fase de còpia per comparativa de dades: Per tal de poder detectar els canvis, necessitem tenir la còpia de les dades del sistema origen d'ahir.

Fase de còpia UXXI a SAKAI: Esborra les taules on tenim la còpia del sistema origen i torna a copiar totes les dades necessàries des de les vistes de UXXI a les taules de SAKAI.

Fase de Baixes: Detecta quins contenidors o usuaris hi ha a la còpia d'ahir i no hi són a la còpia d'avui i els tracta a les taules del Course Management.

Fase de Altes: Detecta quins contenidors o usuaris hi ha no teníem a la còpia d'ahir i sí que hi són a la còpia d'avui i els tracta a les taules del Course Management.

Fase d'Estats: En aquesta fase, es marquen diferents ítems amb estats coneguts per a, posteriorment, poder-ne fer el tractament oportú. Per exemple, marquem assignatures que no tenen encara un espai creat al Campus Virtual per tal que després el treball de Quartz programat crei aquest espai.

5.3.2.2 Implementació del procés de sincronització

Per a la implementació d'aquest procés s'ha escollit l'ús de l'eina Spoon de la suite de Pentaho.

Pentaho es defineix com una plataforma de BI orientada a la solució i centra-

da en processos. La Suite està formada per un conjunt de programes lliures per generar intel·ligència empresarial.

En concret s'ha usat el Pentaho Data Integration, un conjunt d'eines visuals que ens permeten accedir, preparar i combinar les dades evitant codificacions i complexitats ja que les diverses aplicacions incorporen interfícies gràfiques.

L'Spoon és una interfície gràfica d'usuari que permet dissenyar transformacions i treballs que es poden executar mitjançant d'altres eines del Pentaho Data Integration com són Pan i Kitchen.

Pan és un motor de transformació de dades que realitza multitud de funcions com lectura, manipulació i escriptura de dades cap i des de varis orígens.

Kitchen és un programa que executa treballs dissenyats amb Spoon en XML o emmagatzemats en un repositori de base de dades. Normalment els jobs es programen per a ser executats automàticament a intervals regulars.

La següent imatge mostra gràficament totes les fases del procés de sincronització que s'ha introduït en l'apartat anterior:

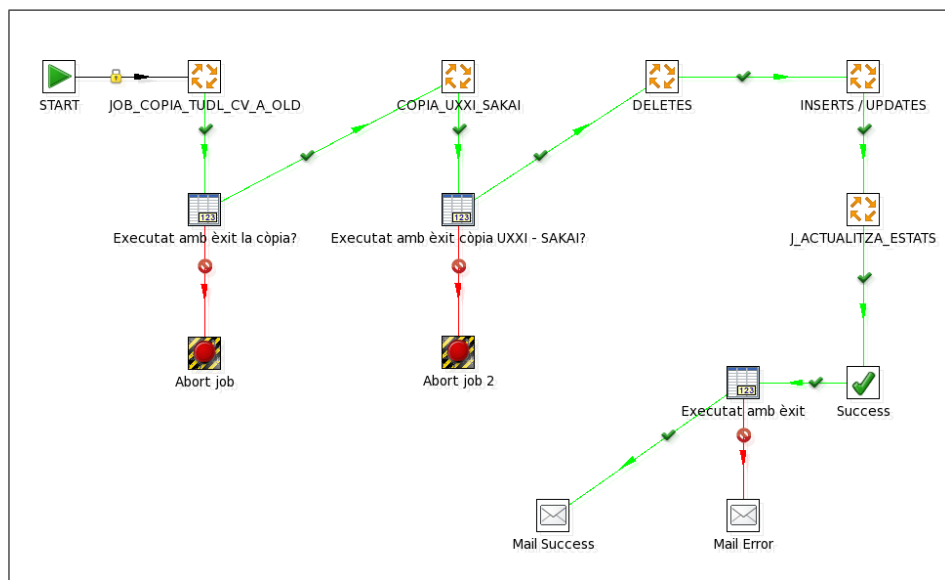


Figura 5.8: Procés de sincronització

60CAPÍTOL 5. COURSE MANAGEMENT: LA IMPLEMENTACIÓ A LA UDL

El treball principal reflexa les 5 fases que acabem de descriure.

En cas d'error tant en la còpia de les taules a una versió anterior com en la còpia de les dades actuals de Uxxi a Sakai, el treball s'atura per tal d'evitar altes o baixes massives en cas que alguna taula hagi quedat buida.

Aquesta comprovació no s'ha fet en el cas de Baixes / Altes, ja que les pròpies transformacions comproven si els registres que calia tractar han estat ja tractats o no. D'aquesta manera, es permet llençar varis cops el treball Deletes o Insert / Updates, si s'hagués donat algun error.

El treball Actualitza Estats, també pot llençar-se varis cops si fes falta, doncs comprova sempre si les dades a actualitzar ja estaven en les corresponents taules d'estats o no.

Els punts més crítics d'aquest treball són els dos primers passos, per això s'hi ha inclòs aquestes comprovacions extres.

Per tal de dur un registre dels canvis que es produeixen cada dia, s'insereix cada acció realitzada a la taula UDL_CM_LOGS.

A continuació s'explica en detall cadascuna de les fases, per les quals, també s'hi contemplen mecanismes de seguretat.

Fase I: Còpia per a comparativa de dades

Com s'ha comentat, és molt important no perdre la referència de les dades de gestió acadèmica d'ahir, per poder després fer les comprovacions dels registres a tractar.

Existeix una taula: UDL_CM_STATUS_JOB , que registra cada execució d'aquest treball i l'estat en què ha acabat. La primera comprovació en començar és mirar en aquesta taula si ja s'ha executat la còpia en el dia d'avui amb èxit. Si és així, no permet continuar amb l'execució, així evitem perdre la referència de les dades.

En cas que les taules de còpia no hi fossin es crearien, i a partir d'aquí copia totes les taules.

Si es produeix un error en qualsevol pas del treball s'envia email d'error.

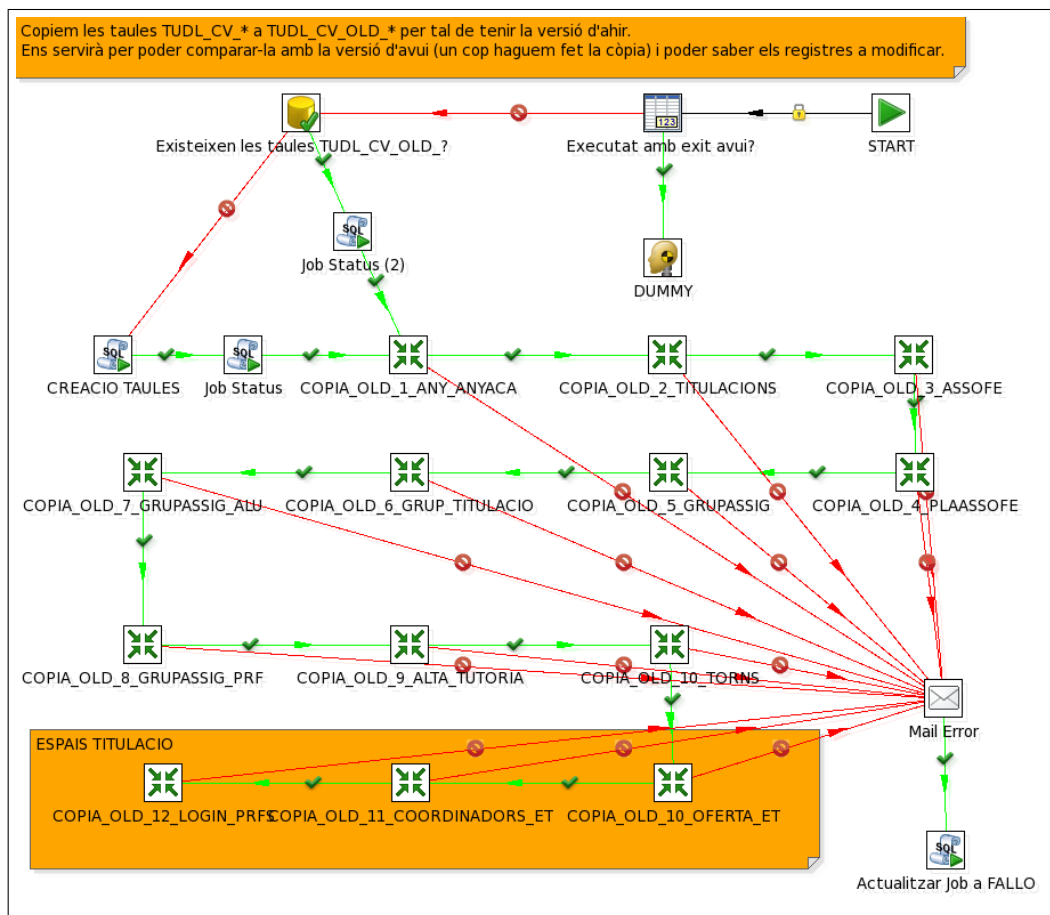


Figura 5.9: Fase I: Còpia de les dades per tenir referència

Fase II: Còpia de UXXI a SAKAI

Aquest treball es fa per temes de rendiment, ja que executar totes les consultes contra les vistes del sistema origen és més costós que contra les taules de la pròpia base de dades de SAKAI, a més correm el risc de perdre'n la connexió en algun moment.

62CAPÍTOL 5. COURSE MANAGEMENT: LA IMPLEMENTACIÓ A LA UDL

La taula UDL_CM_STATUS_JOB registra cada execució d'aquest treball i l'estat en que ha acabat *1 exit*, *0 failure*. La primera comprovació en començar és mirar si ja s'ha executat el treball en el dia d'avui amb èxit. Si és així, no permet continuar amb l'execució.

En cas que les taules de còpia no hi fossin es crearien, i a partir d'aquí copia totes les taules.

Com es detalla en el treball, no es copia la taula TUDL_CV_ANY_ANYACA, ja que d'aquesta manera controlem nosaltres quan canviem d'any al Course Management i en conseqüència al Campus Virtual.

Igual que en la fase anterior, si es produeix un error en qualsevol pas del treball s'envia un email d'error.

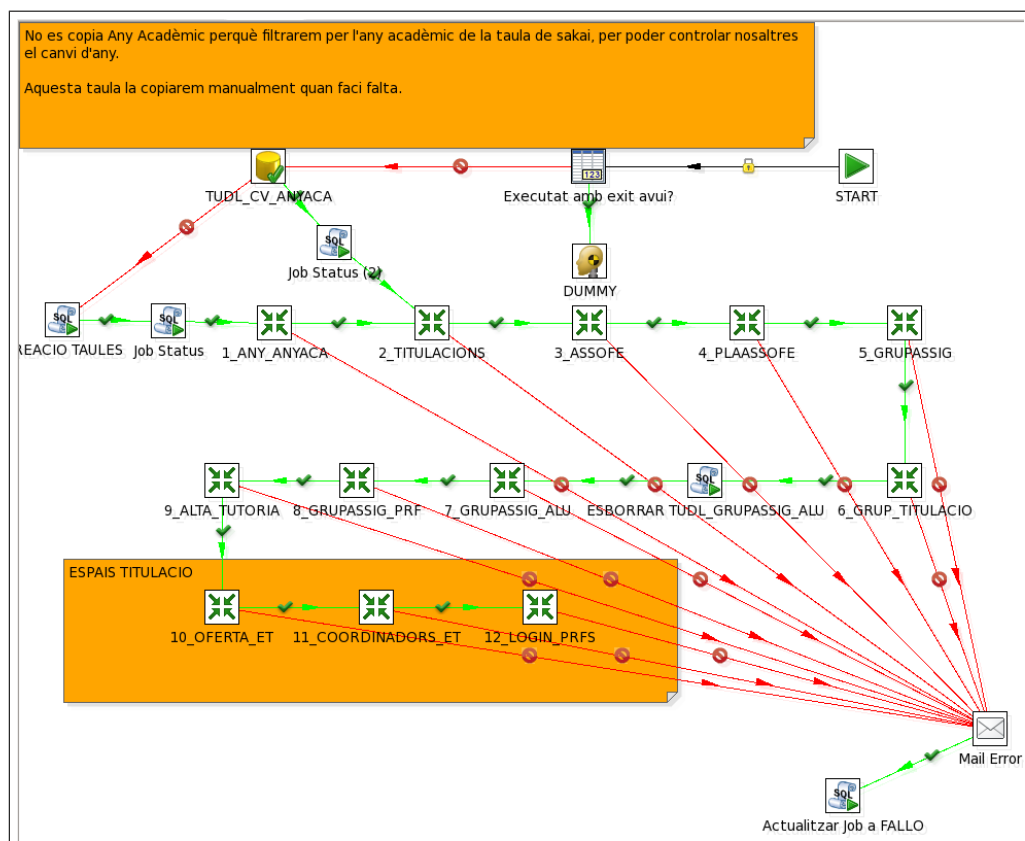


Figura 5.10: Fase II: Còpia de les dades de UXXI a SAKAI

Fase III: Baixes

El Course Management es basa en una estructura jeràrquica de contenidors, en els quals hi podem afegir participants amb diferents rols, a qualsevol nivell de la jerarquia. Per aquest motiu, en el cas de les baixes, cal fer-les de baix cap dalt d'aquesta estructura, doncs no podem esborrar un contenidor en el que encara hi tenim participants.

En tots els passos, es comprova mitjançant la còpia d'ahir i la còpia actual de UXXI els registres a tractar i només s'esborren aquests.

En cas que es produeixi algun error en algun subjob s'envia un correu d'error.

Aquest treball es pot llençar varies vegades seguides, doncs els registres que ja s'han esborrat no els torna a tractar.

Els sub-treballs que integren les baixes són:

- Baixa Alumnes: Esborra alumnes dels grups de matrícula i dels grups de titulació
- Baixa Professors: Esborra professors dels grups de matrícula, de titulació, de Teaching assistants i professors responsables d'assignatures
- Baixa Usuaris genèrics: Esborra els usuaris genèrics (si s'ha esborrat alguna titulació)
- Baixa Oferta: Esborra per ordre grups de titulació, grups de matrícula, grups de TAs, Oferta (pla - ass) i assignatures
- Baixa Espais de Titulació: Esborra per ordre coordinadors, professors, estudiants, grups, oferta i titulacions. En el cas dels espais de titulació CourseOffering = Titulació
- Baixa Titulacions: Esborra les titulacions

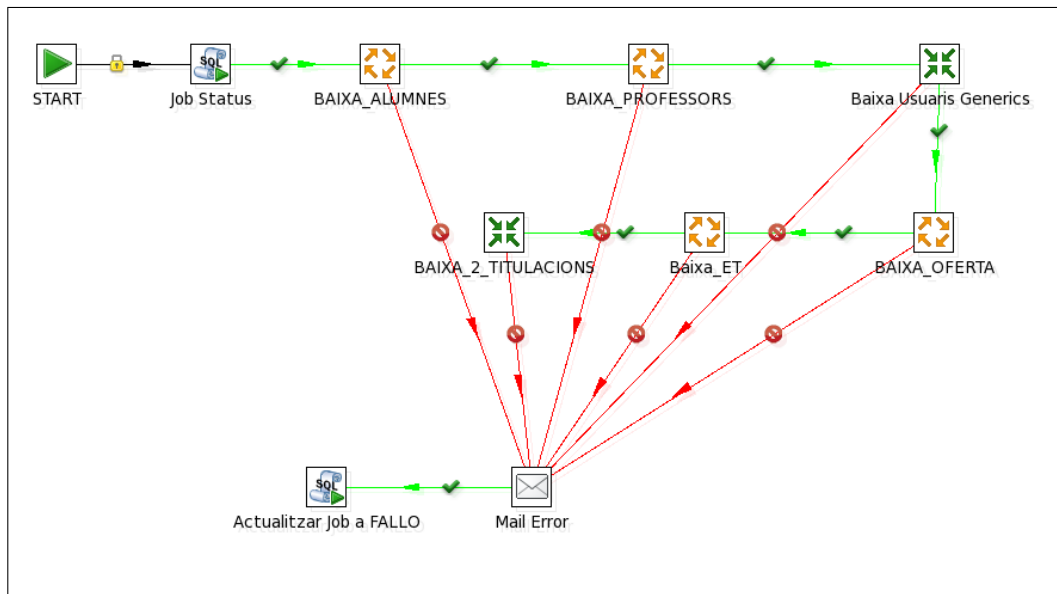


Figura 5.11: Fase III: Tractament de baixes de participants i contenadors

Fase IV: Altes

Com hem comentat, el course management es basa en una estructura jeràrquica de contenidors, en els quals hi podem afegir participants amb diferents rols. Per aquest motiu, en el cas de les altes, cal fer-les de dalt cap baix, doncs no podem inserir un participant en un contenidor que encara no hem creat.

Aquest treball, a més de les altes també tracta modificacions, per exemple: canvi de nom de un grup, canvi de crèdits de un alumne, ...

En tots els passos, es comprova mitjançant la còpia d'ahir i la còpia actual de UXXI els registres a tractar i només tractem aquests, a més s'inclou un pas en cada transformació per comprovar si el registre que anem a tractar ja l'hem tractat abans, per exemple si ja existeix a la base de dades un usuari en un grup que calia donar d'alta, no el torna a tractar. Gràcies a aquestes comprovacions, en cas que calgui executar aquest treball algun altre cop el mateix dia, per exemple si ha donat algun error, es pot fer sense problema.

En cas que en algun subjobs es produeixi un error s'envia un correu d'error.

Els subjobs que integren les altes i modificacions són:

- Oferta: Afegeix o modifica per ordre, academic sessions, titulacions, assignatures, oferta, grups de matrícula, grups de titulació i grups de Teaching Assistants (professors de pràctiques).
- Professors: Afegeix o modifica professors de matrícula, de titulació, teaching assistants o professors responsables d'assignatures
- Alumnes: Tracta l'alta o modificació d'alumnes de grups de matrícula i alumnes de grups de titulació.
- Espais de Titulació: Tracta per ordre, titulacions ET (en el cas de ET -*espais de titulació*- són courseOfferings), oferta ET, grups ET, estudiants ET, professors ET i coordinadors ET.
- Usuaris genèrics: En el cas que s'hagi creat una nova titulació, s'afegirà l'usuari genèric.

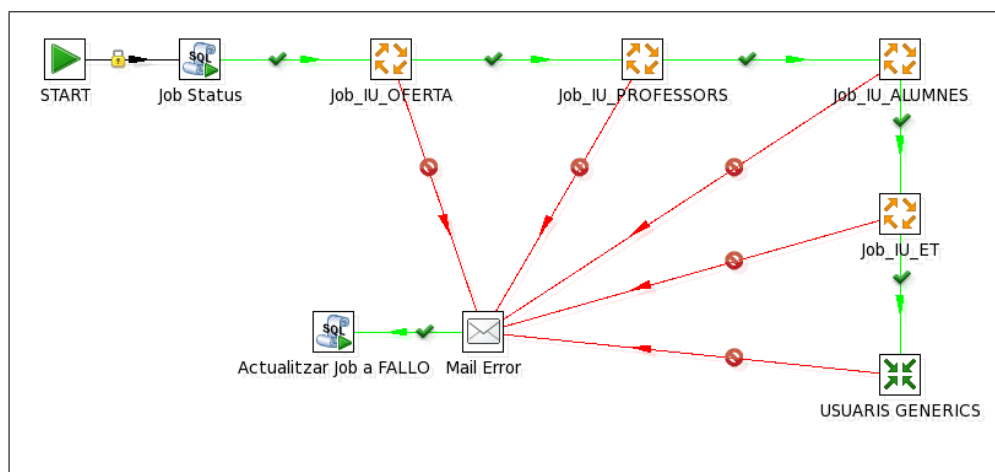


Figura 5.12: Fase IV: Tractament de les altes de participants i contenadors

Fase V: Estats

Aquest treball, tracta els diferents ítems que cal marcar per al seu posterior tractament. A continuació es detalla els ítems que necessitem tenir identificats així com el treball de Quartz implementat per a tractar-lo.

- ACTUALITZA ESTATS ASSIGNATURES: Detecta assignatures planificades que no tenen el seu corresponent espai de docència creat al Campus Virtual i les insereix a la taula corresponent. Cada nit s'executa el job de Quartz **courseManagementSincro**, que llegeix aquestes assignatures de la taula d'estats corresponent i crea al Campus Virtual un espai nou de tipus course a partir de la plantillaCurs.
- ACTUALITZA ESTAT EXTERNS: Tracta els professors externs detectats que no tenen un usuari creat al Campus Virtual, inserint-los a la taula d'estats corresponents. Cada nit s'executa el job de Quartz **usuarisExternsSincro**, que crea un usuari al Campus per cada usuari extern i els hi envia un mail per poder activar-se el compte.
- ACTUALITZA ESTATS ET: Insereix a la taula d'estats corresponent els nous espais de titulació que no tenen el seu espai corresponent creat al Campus Virtual. Cada nit s'executa el job de Quartz **espaisTitulacioSincro** que crea un espai de tipus course ja que ve proveït per Course Management, mitjançant la plantillaET.
- ACTUALITZA ESTATS PAT: Detecta els alumnes nous de primer amb el torn que fan. Cada nit s'executa un job Quartz **altaAlumnesPAT** que els assigna un grup aleatori de tutoria.
- ACTUALITZA ESTATS GRUPS NOUS: Quan detecta una alta de un grup nou per a un espai d'assignatura que ja està creat, es busca el professor

responsable de l'assignatura a la que pertany el grup nou i el seu correu electrònic i s'insereix a la taula d'estats corresponent. Cada nit s'executarà un job de Quartz **notificacioGrupsNous**, que envia als professors inserits un avís indicant que tenen disponible un nou grup d'assignatura i les instruccions per incloure'l en l'espai que desitgin.

Cal tenir molt en compte no tenir activat aquest job durant els primers dies de càrrega de assignatures al Campus quan encetem un nou curs.

- **ACTUALITZA REALMS A UPDATER:** Comprova en quins contenidors s'ha produït canvis i els afegeix a una taula d'estats que posteriorment consultarà el job de Quartz que actualitza els realms.

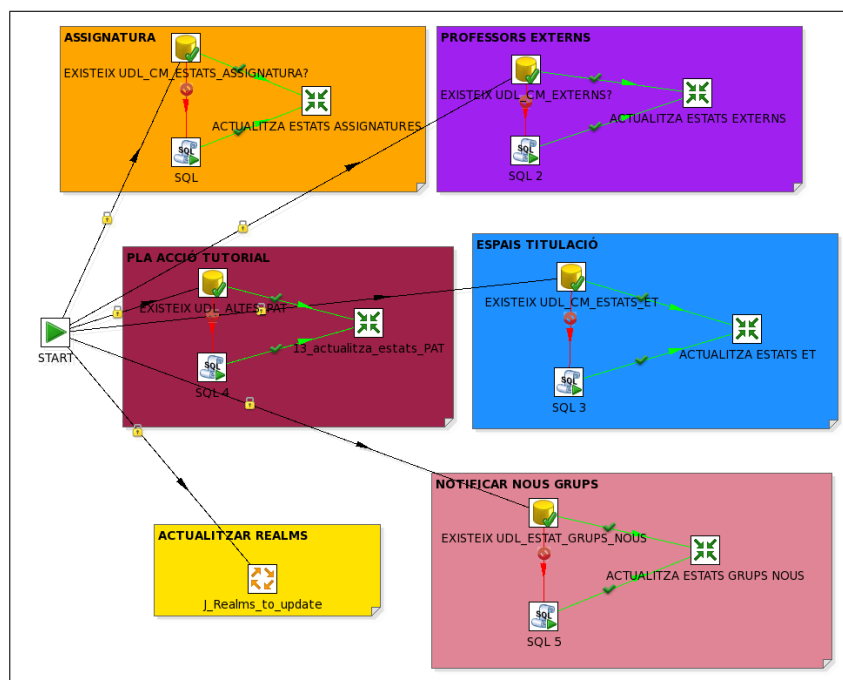


Figura 5.13: Fase V: Actualitza els estats de diferents elements

En l'annex A s'especifica en detall els fitxers inclosos en cadascun d'aquests directoris, i el cd adjunt a aquest projecte inclou tots els fitxers.

5.3.3 Jobs de Quartz

Quartz és una completa llibreria de codi lliure per a la programació de tasques, que pot ser integrada o usada amb qualsevol aplicació J2EE o J2SE. Quartz es pot emprar per a crear programacions simples o complexes per a executar desenes, centenars o milers de tasques. Aquestes tasques es defineixen com a components estàndards de Java o EJBs.

Aquesta llibreria ja està integrada a Sakai des de la versió 2.3.

Aprofitant aquesta llibreria, s'ha desenvolupat totes les tasques necessàries per a la gestió del Course Management. La implementació detallada d'aquestes tasques es pot consultar en l'ANNEX A.

1. Actualitzar els realms dels espais connectats amb el Course Management

La informació sobre participants, rols i sections a les que pertanyen es desa als realms dels espais. Aquests realms s'actualitzen quan es desen, per aquest motiu, si hi ha canvis respecte els participants d'una section o dels rols que tenen assignats, no es veuran reflectits fins que no es desi el realm.

Cal doncs actualitzar aquests realms dels espai o grups que estan integrats amb el Course Management. La implementació de referència del Course Management inclou un job de Quartz que fa aquesta actualització, no obstant decidim modificar-lo per tal que actualitzi només els realms dels espais que han sofert alguna modificació agilint-ne així l'execució.

Aquest job s'executa diàriament.

2. Creació dels espais d'assignatura

Aquesta tasca consulta la taula d'estats d'assignatura que s'ha detallat en l'apartat anterior, i crea un espai de tipus course per a les assignatures que s'ha detectat que encara no tenen el seu espai associat. Alhora, també els afegeix un proveïdor amb l'identificador de tots els grups de matrícula que componen l'assignatura, gràcies al qual s'omplirà el realm de l'espai i dels grups.

3. Creació dels espais de titulació

Aquesta tasca consulta la taula d'estats dels espais de titulació detallada en l'apartat anterior, i crea un espai de tipus course per a cada espai de titulació que encara no en té .

En el cas dels espais de titulació, els grups que es creen equivalen a cadascuna de les assignatures planificades per aquella titulació concreta.

4. Creació dels professors externs

Donat que es permet la planificació de professors externs, hi ha una transformació inclosa en el procés de sincronització de Pentaho que detecta aquells professors que no tenen compte a la UdL i els inclou a la taula d'estats corresponent.

La tasca de creació d'aquests usuaris externs, crea un usuari al Campus Virtual, de manera que en rebre'l pot validar el compte seguint l'enllaç que s'hi indica.

5. Notificació grups nous planificats

Un cop s'ha creat els espais de tipus course i s'hi ha afegit els proveïdors, si es planifica un nou grup en qualsevol assignatura, la tasca de creació d'assignatures ja no ho detecta, doncs l'espai ja està creat.

Aprofitant que la integració amb el Course Management dona al professor molta llibertat en la configuració i gestió dels seus espais, es va decidir no modificar espais que ja s'havien creat, de manera que quan es detecta la planificació d'un nou grup d'assignatura, aquesta tasca, envia un correu electrònic a la persona responsable de l'assignatura i és aquesta qui decideix si incorporar el grup a l'espai o no.

En l'annex A s'especifica en detall els fitxers inclosos en cadascun d'aquests directoris, i el cd adjunt a aquest projecte inclou tots els fitxers.

5.4 Test

Una fase molt important en tot projecte és la fase de test, cal comprovar d'una banda que les noves funcionalitats que s'han implementat funcionen correctament, però també que tot allò que pugui veure's afectat per aquestes noves funcionalitats, continuen comportant-se de la forma esperada.

El Course Management aprovisionarà la informació necessària per als espais de tipus curs.

Pel que fa a la creació d'espais, hi haurà espais que es crearan automàticament i d'altres que la seua creació es deixarà a la decisió de l'usuari.

Pel que fa a l'assignació dels participants als espais amb el rol corresponent, serà el proveïdor del Course Management qui ho gestionarà, però és en la creació dels espais que assignarem els proveïdors corresponents a cada espai.

El course management en sí, no és una eina, de manera que per comprovar-ne el bon funcionament, hem de testejar les eines que consumeixen aquestes dades que aprovisiona.

En l'elaboració d'aquest pla de test, s'ha comprovat el funcionament de totes les eines del Campus Virtual, i s'han escrit els tests pertinents, en cas que l'eina concreta estigui o pugui estar integrada amb el Course Management.

5.4.1 Pla de test

S'ha dividit els tests en:

- Tests dels treballs de sincronització: Han servit per comprovar que totes les transformacions programades amb l'eina Spoon funcionen correctament.
- Tests dels treballs de Quartz per a la creació d'espais al Campus Virtual: Han servit per a comprovar el funcionament dels diferents treballs de Quartz programats.
- Tests de funcionalitats de les eines que poden integrar-se amb el Course Management: Amb aquests tests s'ha comprovat les diferents funcionalitats de les eines que fan ús del Course Management.

Per tal de poder passar tots aquests tests, s'ha disposat d'un servidor de proves amb un Campus Virtual desplegat incloent tota la implementació referent al Course Management, i d'una base de dades de proves.

72CAPÍTOL 5. COURSE MANAGEMENT: LA IMPLEMENTACIÓ A LA UDL

Aquests tests els han passat diferents programadors de l'equip per separat. A continuació es mostra un exemple de test concret:

The screenshot shows a Confluence page for a test titled "TINI3 - Professor entra al Worksite Setup i té a la seva disposició totes les assignatures i grups als que està planificat + Grups Titulació + Grups TA". The page is structured as follows:

- Title:** TINI3 - Professor entra al Worksite Setup i té a la seva disposició totes les assignatures i grups als que està planificat + Grups Titulació + Grups TA
- Description:** Aquest test ens servirà per comprovar que les dades que hem importat d'UXXI s'han convertit correctament a les taules del Course Management i que cada perfil té accés a les dades correctes.
- Objectius:**
 - Comprovar que es veuen les dades correctes per cada perfil
- Procediment:**
 - Entrar amb l'usuari detallat a l'annex al **Campus Virtual Real**.
 - Anar a l'eina Configuració de l'espai
 - Rol Student -> Pot fer click al botó 'Nou', però només pot crear espais de tipus Projecte o Portfoli.
 - Rol Teaching Assistant -> Pot fer click al botó 'Nou', però només pot crear espais de tipus Projecte o Portfoli, o bé pot sol·licitar a l'administrador la creació d'espais proveïts de docència reglada
 - Rol Instructor -> Pot crear espais de qualsevol tipus. Quan va a crear un espai de tipus curs, li apareixen tots els grups pels quals és Instructor.
 - Rol CourseAdmin -> Pot crear espais de qualsevol tipus. Quan va a crear un espai de tipus curs, li apareixen tots els grups d'aquelles assignatures (CourseOffering) en els quals té aquest rol. **PREGUNTA** ok
- Resultats obtinguts:**
 - Supera el test? SI / NO
 - Observacions:

A l'entrar amb un usuari (afs2) amb rol estudiant no apareix el botó **Nou** a Configuració d'espais. Li apareix la llista d'espais de la qual és membre **PREGUNTA** ok

Figura 5.14: Exemple de test de funcionalitat

Capítol 6

Metodologia i eines emprades

Aquest projecte va néixer amb un objectiu molt clar i definit: crear una implementació de la API del Course Management de Sakai.

Malgrat tenir aquest objectiu clar, la forma d'assolir-lo no era gens clara: el desconeixement a priori de la API a implementar feia molt difícil predir quin era l'abast del projecte i les tasques concretes que s'haurien de dur a terme. La seva evolució, per tant, era difícil de predir i planificar. Justament per aquest motiu no era gens aconsellable utilitzar en aquest projecte metodologies tradicionals de gestió de projectes en el desenvolupament de programari, i es va considerar més oportú i òptim emprar una metodologia àgil que permetés adaptar la seva planificació a l'evolució en l'aprenentatge respecte l'abast real del projecte.

Les metodologies tradicionals recomanen gestionar un projecte per fases (planificació, anàlisi, disseny, implementació) que es succeeixen de forma seqüencial, servint el resultat d'una fase com a punt de partida de la següent. Per això aquestes disciplines fan èmfasi en la planificació de la totalitat del treball a realitzar, i un cop detallat, comença el cicle de desenvolupament del software. Poden aplicar-se en projectes en què es coneix molt bé des de l'inici l'abast i les tasques necessàries

per portar-lo endavant. El seu major inconvenient és que no s'adapten bé als canvis, donat que un canvi detectat en la fase de disseny obligaria a tornar a la fase de planificació.

Les metodologies àgils van néixer davant la constatació que la majoria de projectes de desenvolupament són extremadament propensos a patir canvis durant la seva construcció, sobretot en casos en que es tracta de crear noves solucions a problemes no exhaustivament definits.

Per aquest motiu, les metodologies àgils recomanen crear solucions de forma iterativa i incremental. El projecte es divideix en iteracions amb una durada fixa determinada. Cada iteració té com a objectiu construir un conjunt mínim de funcionalitats, i en cada iteració es procedeix en fases d'anàlisi, disseny i implementació.

L'objectiu principal d'aquest tipus de metodologies és mostrar resultats a l'usuari final de forma ràpida, per tal de recollir la seva opinió i esbrinar, el més aviat possible, si vol fer canvis.

DIFERENCIAS ENTRE METODOLOGÍA TRADICIONALES Y ÁGILES	
Metodologías Tradicionales	Metodologías Ágiles
Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo	Basadas en heurísticas provenientes de prácticas de producción de código
Cierta resistencia a los cambios	Especialmente preparados para cambios durante el proyecto
Impuestas externamente	Impuestas internamente (por el equipo)
Proceso mucho más controlado, con numerosas políticas/normas	Proceso menos controlado, con pocos principios.
El cliente interactúa con el equipo de desarrollo mediante reuniones	El cliente es parte del equipo de desarrollo
Más artefactos	Pocos artefactos
Más roles	Pocos roles
Grupos grandes y posiblemente distribuidos	Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio
La arquitectura del software es esencial y se expresa mediante modelos	Menos énfasis en la arquitectura del software
Existe un contrato prefijado	No existe contrato tradicional o al menos es bastante flexible

Figura 6.1: Metodología tradicional vs Metodología àgil

En aquest projecte, s'ha seguit una metodologia àgil en el sentit de planificar amb iteracions de durada determinada, de detectar possibles canvis el més aviat possible, d'integrar contínuament el projecte a la resta de codi per detectar possibles errors, d'emprar eines de treball col·laboratiu, ... tot i que estrictament, no teníem un usuari o client final a qui mostrar-li resultats, sinó que hem estat nosaltres mateixos els que hem anat planificant, marcant objectius i refinant-los a mesura que avançava el projecte.

A continuació es mostra, el calendari d'iteracions que es va marcar inicialment, les quals s'han anat refinant, reestructurant i modificant al llarg del projecte. Inicialment es va marcar 15 dies per cada iteració:

<input checked="" type="checkbox"/>	CM-1	Estudi del CourseManagement
<input checked="" type="checkbox"/>	CM-2	Desenvolupament d'un mockup per agafar habilitats amb l'api
<input checked="" type="checkbox"/>	CM-3	Importar documentació tipologia espais
<input checked="" type="checkbox"/>	CM-4	Implementació del proveïdor de cursos
<input checked="" type="checkbox"/>	CM-5	Documentació nou model de course management
<input type="checkbox"/>	CM-6	Recollida de requeriments de course management
<input checked="" type="checkbox"/>	CM-7	Estudi del procés automatitzat d'assignatures i associació de sections
<input checked="" type="checkbox"/>	CM-8	Presentació del mecanisme a membres d'ASIC de GA
<input checked="" type="checkbox"/>	CM-9	Procés traspàs de dades UXXI - SAKAI

Figura 6.2: Definició inicial de les iteracions del projecte

6.1 Metodologia de desenvolupament

La subàrea de gestió de projectes de campus de l'Àrea de Sistemes d'Informació i Comunicació de la UdL som un equip de desenvolupadors que programem sovint contra la mateixa plataforma. Els requeriments canviants de les diferents aplicacions, la compartició de recursos (servidors, repositoris, ...) i la variabilitat en les tasques que realitzem fan necessària una gestió de projectes, i per la naturalesa d'aquests, l'ús de metodologies àgils ens permeten dur a terme un treball col·laboratiu, amb integració contínua i control de versions d'una manera eficient.

És per aquest motiu, juntament amb la resta de motius exposats en l'apartat anterior que la planificació, estructuració i desenvolupament d'aquest treball s'ha dut a terme seguint els principis del manifest àgil.

Com s'ha comentat som un equip desenvolupant sota la mateixa plataforma, noves funcionalitats del Campus Virtual, resolució de bugs, noves eines, ... I donant servei del Campus Virtual 24x7. Per tal d'organitzar-nos en el nostre treball diari, disposem de 3 entorns de desenvolupament:

- **devel:** Servidor amb un entorn de desenvolupament. Cada programador es descarrega una còpia del codi en local, i es la que fa servir per a realitzar les seves tasques.
- **preproducció:** Entorn on integrem tots els canvis que realitzem, es tracta d'un entorn idèntic a producció.
- **producció:** Entorn real, qualsevol canvi que s'hagi d'aplicar en aquest entorn es realitza sempre per la nit. I només si la integració a preproducció s'ha realitzat amb èxit.

Disposem també d'un repositori, per emmagatzemar i versionar el codi que escrivim, per cada entorn de desenvolupament on hi distingim dos branques: la branca de desenvolupament (*devel*) i la branca de producció (*prod*).

En primer lloc, quan un programador finalitza el desenvolupament d'una tasca en el seu entorn local, puja els canvis a l'entorn de *devel*. Mijtançant una eina d'integració contínua¹, quan es detecten canvis en el codi, es procedeix a compilar-lo i desplegar-lo de nou, de manera, que podem comprovar en primera instància si els canvis que hem realitzat s'han aplicat bé al servidor de desenvolupament.

Un cop aquests canvis ja funcionen a *devel* i s'ha passat els tests d'acceptació, continuem amb la integració a preproducció per tal de comprovar que no només els canvis s'apliquen bé, sinó que no interfereixen en el funcionament de la resta de codi i que en un entorn idèntic al de producció el comportament és l'esperat. Per tal d'integrar els canvis a preproducció cal aplicar els canvis a la branca *prod* del repositori. L'eina que utilitzem per a la integració contínua, en detectar canvis a la branca de producció inicia el procés de compilar i desplegar el nou codi en el servidor de preproducció.

En tots els casos, la mateixa eina d'integració continua passa, un cop compilat el codi, tots els tests unitaris que hem escrit, de manera que les funcionalitats bàsiques que volem provar amb aquests tests ja es realitzen de forma automàtica.

Si en l'entorn de preproducció es detecta qualsevol error, ja sigui de compilació, de tests unitaris, o bé perquè ens adonem que el resultat final en aquest entorn no es comporta com esperàvem i cal modificar-lo, s'atura el procés i cal que apliquem manualment una versió anterior de codi per tal d'evitar que els canvis s'apliquin també a l'entorn de producció.

¹Una eina d'integració contínua ens permet prevenir i reduir els errors en el pas a producció.

Si en canvi, tot funciona perfectament, els canvis a producció s'aplicaran a la nit. L'eina d'integració contínua està programada per aturar els servidors de producció escalonadament, és a dir, un després de l'altre, per tal de no deixar de donar servei de Campus Virtual. I en cada servidor, s'aplicaran els nous canvis.

A continuació es mostra de forma gràfica tot el descrit fins ara:

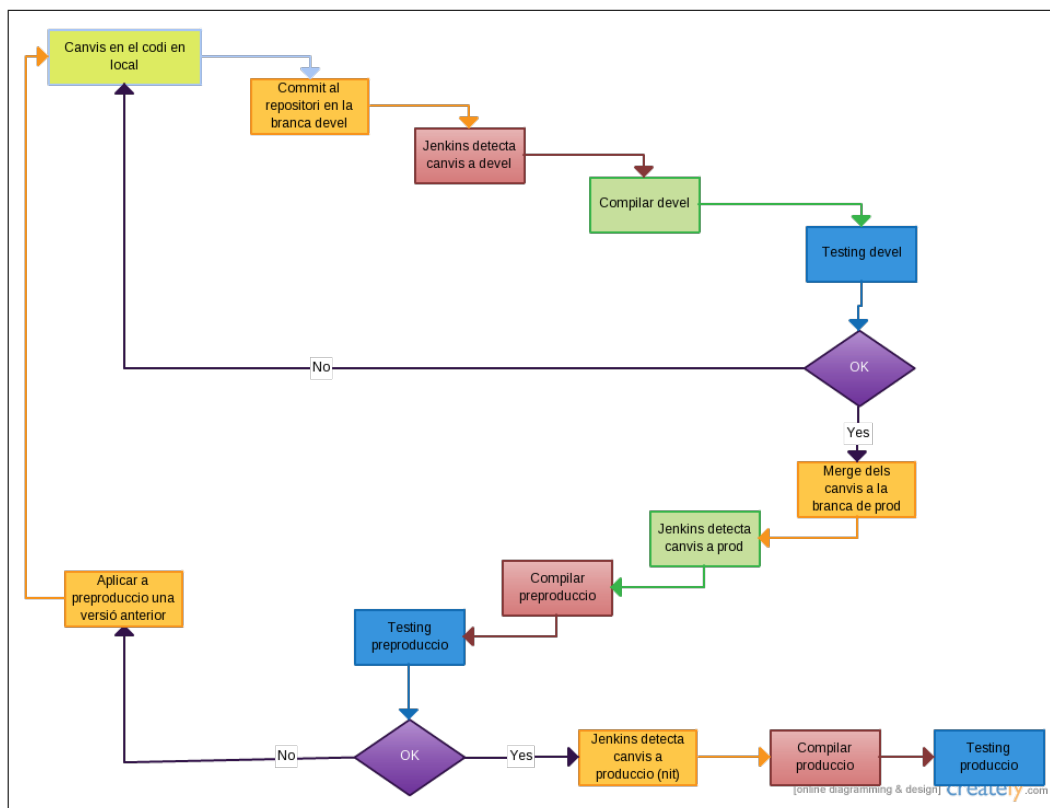


Figura 6.3: Diagrama de flux integració de codi

6.2 Eines emprades

Per tal de dur a terme tota la metodologia de treball descrita en l'apartat anterior, utilitzem les següents eines:

Subversion Com a eina de control de versions open source basat en un repositori, el funcionament del qual s'assembla bastant al d'un sistema de fitxers. Utilitza el concepte de revisió per a desar els canvis produïts en el repositori. Entre dos revisions només desa el conjunt de modificacions, optimitzant així l'ús d'espai en disc. Fomenta el treball col·laboratiu, doncs permet que varies persones puguin modificar i administrar el mateix conjunt de dades des de les seves respectives ubicacions.

El repositori està centralitzat en un servidor, i mitjançant comandes bàsiques des del client podem interactuar amb el servidor per a realitzar les accions desitjades (check-out, commit, update, delete, ...)



Figura 6.4: Revisió de codi amb Subversion

Jenkins És un software d'integració continua open source escrit en Java, que tenim centralitzat en un servidor. Proporciona integració continua per al desenvolupament de software. Suporta eines de control de versions com CSV, Subversion, Git, ... i pot executar projectes basats en Apache Ant i Apache Maven.

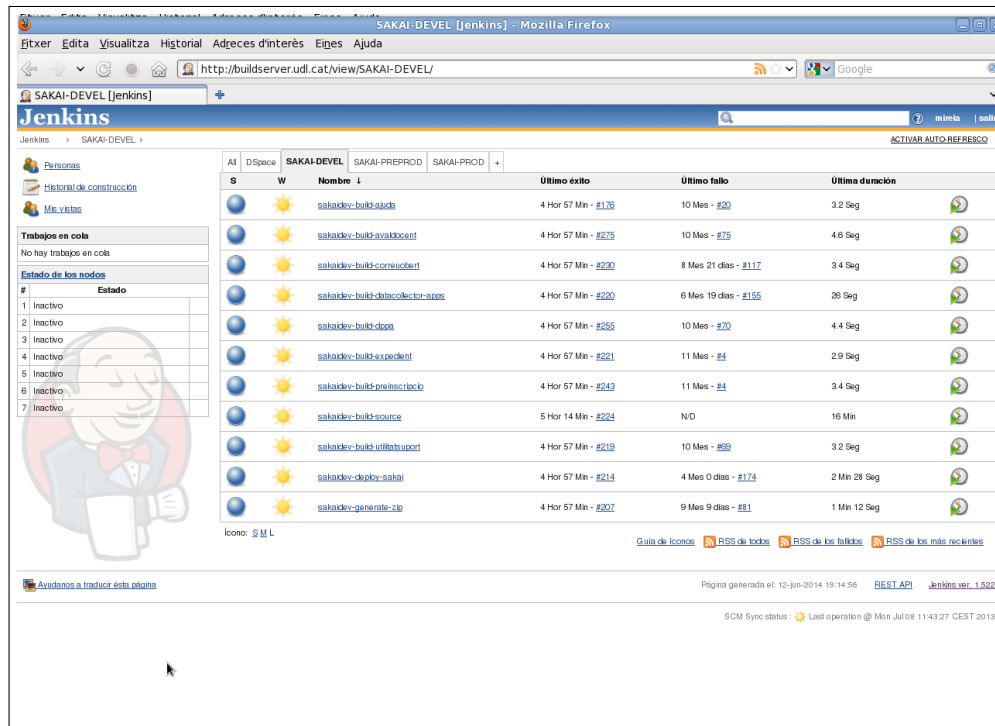


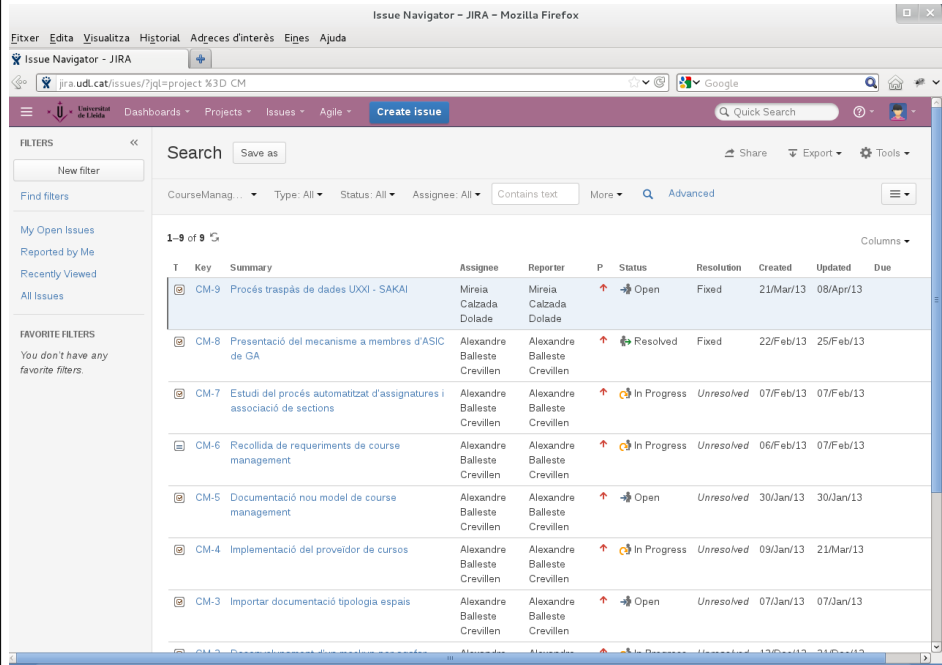
Figura 6.5: Exemple de tasques planificades per als diferents entorns amb Jenkins

Maven És una eina de software per la gestió i construcció de projectes Java. És similar en funcionalitat a Apache Ant, però té un model de construcció més simple, basat en un format XML. Maven utilitza un Project Object Model (POM) per a descriure el projecte de software a construir, les seves dependències d'altres mòduls i components externs i l'ordre de construcció dels elements. Realitza tasques de compilació del codi, execució de tests unitaris i d'empaquetat, i a més, ens serveix també per a versionar els nostres propis paquets i poder conèixer així els canvis en cada versió.

Jira És una aplicació basada en web que pot ser utilitzada per a la gestió operativa de projectes, per a la gestió de processos i per la millora de processos. Jira permet:

- Capturar i organitzar el treball de l'equip, prioritzar i actuar sobre el que és realment important i estar al dia del que passa.
- Proporcionar fluxos de treball que s'ajusten als processos existents i que es poden adaptar alhora que l'equip evoluciona.
- Fer un seguiment del que ja està fet amb facilitat.
- Seguir les tasques més importants, monitoritzar els fluxos d'activitat i compartir informació.

A més existeix un plugin de Subversion per a Jira, la qual cosa ens permet lligar la tasca planificada amb el codi relacionat.



T	Key	Summary	Assignee	Reporter	P	Status	Resolution	Created	Updated	Due
CM-9	CM-9	Procés traspàs de dades UXXI - SAKAI	Mireia Calzada Dolade	Mireia Calzada Dolade	↑	Open	Fixed	21/Mar/13	08/Apr/13	
CM-8	CM-8	Presentació del mecanisme a membres d'ASIC de GA	Alexandre Ballesta Crevillen	Alexandre Ballesta Crevillen	↑	Resolved	Fixed	22/Feb/13	25/Feb/13	
CM-7	CM-7	Estudi del procés automatitzat d'assignatures i associació de seccions	Alexandre Ballesta Crevillen	Alexandre Ballesta Crevillen	↑	In Progress	Unresolved	07/Feb/13	07/Feb/13	
CM-6	CM-6	Recollida de requeriments de course management	Alexandre Ballesta Crevillen	Alexandre Ballesta Crevillen	↑	In Progress	Unresolved	06/Feb/13	07/Feb/13	
CM-5	CM-5	Documentació nou model de course management	Alexandre Ballesta Crevillen	Alexandre Ballesta Crevillen	↑	Open	Unresolved	30/Jan/13	30/Jan/13	
CM-4	CM-4	Implementació del proveïdor de cursos	Alexandre Ballesta Crevillen	Alexandre Ballesta Crevillen	↑	In Progress	Unresolved	09/Jan/13	21/Mar/13	
CM-3	CM-3	Importar documentació tipologia espais	Alexandre Ballesta Crevillen	Alexandre Ballesta Crevillen	↑	Open	Unresolved	07/Jan/13	07/Jan/13	

Figura 6.6: Exemple de tasques definides per al Projecte Course Management

A continuació es mostra el detall d'una issue concreta:

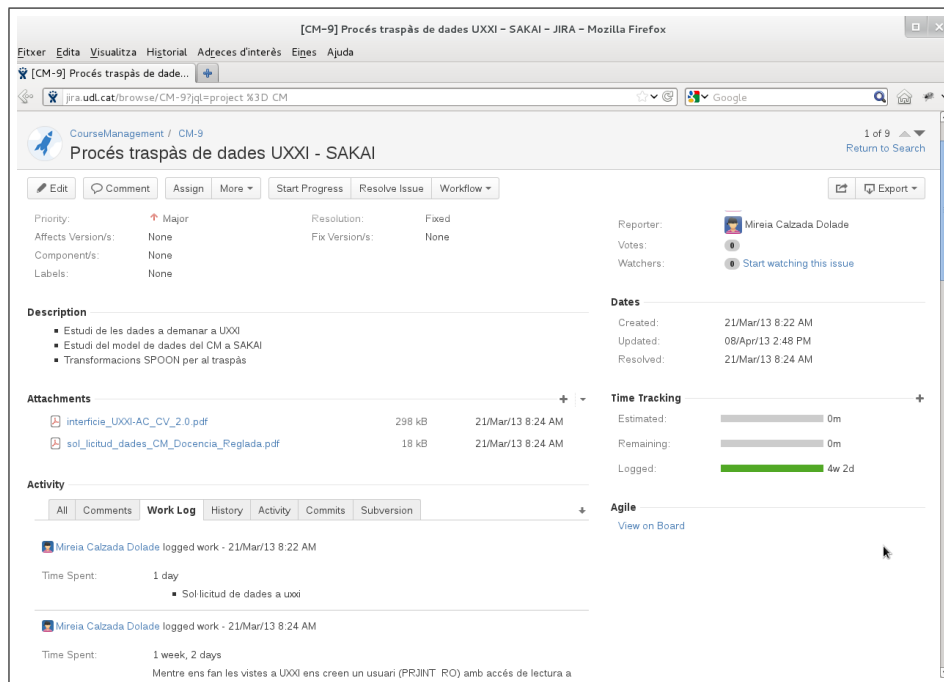


Figura 6.7: Tasca concreta de Jira

Confluence És una eina de treball col·laboratiu que permet organitzar la informació per projectes, per departaments o per equips, alhora que la manté ordenada i accessible mitjançant una jerarquia flexible de la pàgina.

Ens permet compartir qualsevol tipus d'informació, ja sigui notes de reunions, requisits de productes, planificacions de projectes i també manté versions de fitxers.

Promou el debat, doncs permet fer comentaris a les diferents publicacions de l'equip.

Està integrat amb Jira, de manera que relaciona les tasques planificades en Jira amb la documentació associada al Confluence.

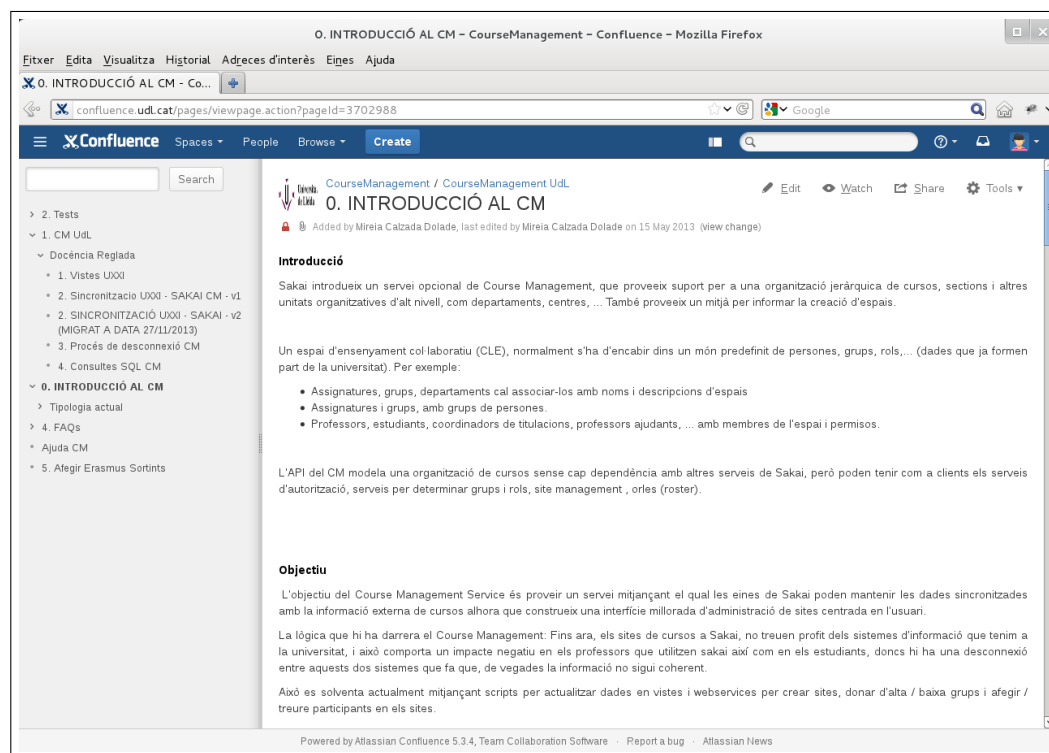


Figura 6.8: Exemple de documentació escrita al Confluence

Totes les eines exposades fins ara, les utilitzem en el nostre dia a dia com a metodologia de treball. Per al desenvolupament d'aquest projecte concret, també s'ha usat de manera local les següents:

Eclipse És un entorn integrat de desenvolupament, és a dir, un conjunt d'utilitats per a crear programari (editor de codi font, compilador, depurador) agrupades en un mateix entorn. Eclipse està fet en Java i pensat principalment per a crear aplicacions en Java, tot i que també es pot utilitzar per a crear aplicacions en altres llenguatges.

A banda de les utilitats típiques de qualsevol IDE (Entorn integrat de desenvolupament), Eclipse té funcionalitats molt interessants com:

- Integració amb Maven
- Integració Subversion
- Extensions de treball amb bases de dades
- Extensió d'integració amb servidors d'aplicacions (Tomcat, BEA ...)
- Extensions de creació de aplicacions J2EE

Funciona amb perspectives (conjunt de vistes) i vistes.

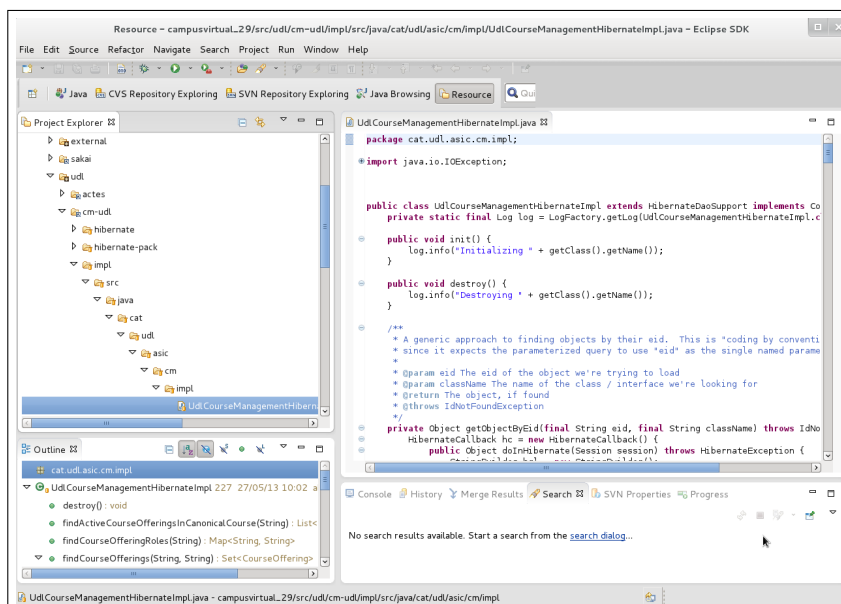


Figura 6.9: Finestres de treball en Eclipse

Oracle Sqldeveloper És l'eina gràfica que proporciona Oracle per a treballar amb bases de dades, permet desenvolupar i executar consultes o scripts SQL, tant DML com DDL sobre bases de dades Oracle.

Permet connectar també amb d'altres bases de dades que no siguin Oracle, com MySQL, SQLServer o Access.

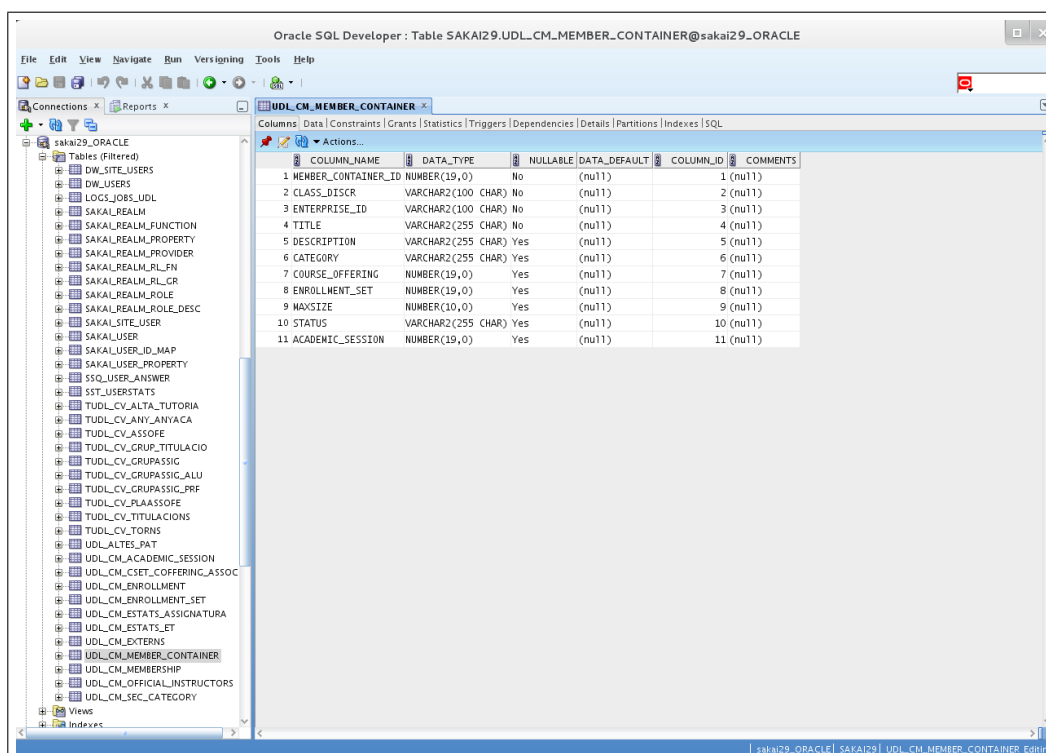


Figura 6.10: Finestres de treball del Sqldeveloper

Pentaho Pentaho es defineix com una plataforma de BI orientada a la solució i centrada en processos. La Suite està formada per un conjunt de programes lliures per generar intel·ligència empresarial.

En concret, en aquest treball s'ha usat el Pentaho Data Integration, un conjunt d'eines visuals que ens permeten accedir, preparar i combinar les dades evitant codificacions i complexitats ja que les diverses aplicacions incorporen interfícies gràfiques.

L'Spoon és una interfície gràfica d'usuari que permet dissenyar transformacions i treballs que es poden executar mitjançant d'altres eines del Pentaho Data Integration com són Pan i Kitchen.

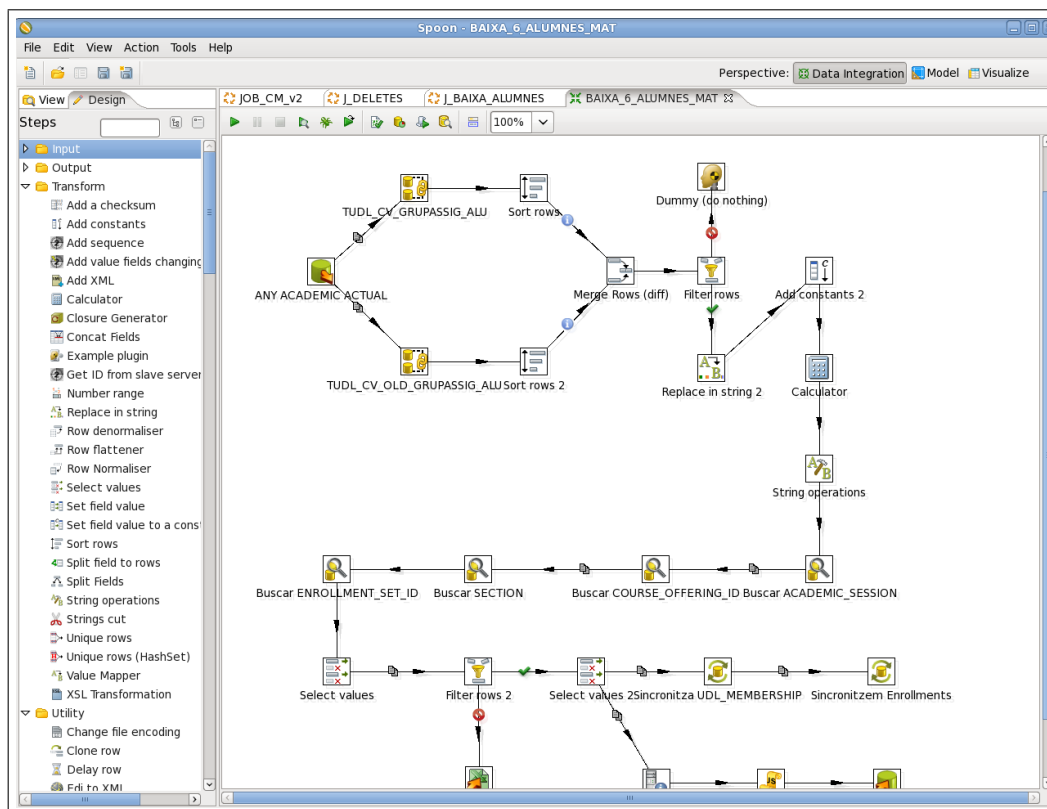


Figura 6.11: Finestres de treball d'Spoon

Pan és un motor de transformació de dades que realitza multitud de funcions com lectura, manipulació i escriptura de dades cap i des de varis orígens.

Kitchen és un programa que executa treballs dissenyats amb Spoon en XML o emmagatzemats en un repositori de base de dades. Normalment els treballs es programen per a ser executats automàticament a intervals regulars.

De la mateixa manera, que tot el codi que generem a la nostra subàrea es desa en un repositori versionat amb Subversion, totes les transformacions i tre-

balls que s'han generat amb l'Spoon per tal de dur a terme la sincronització de dades amb els sistemes origen també les desem i versionem.

Anàlogament també hem creat un entorn de desenvolupament i un entorn de producció, doncs tots els canvis que provoquen aquestes transformacions afecten les bases de dades, i evidentment cal integrar-los abans de passar-los a producció.

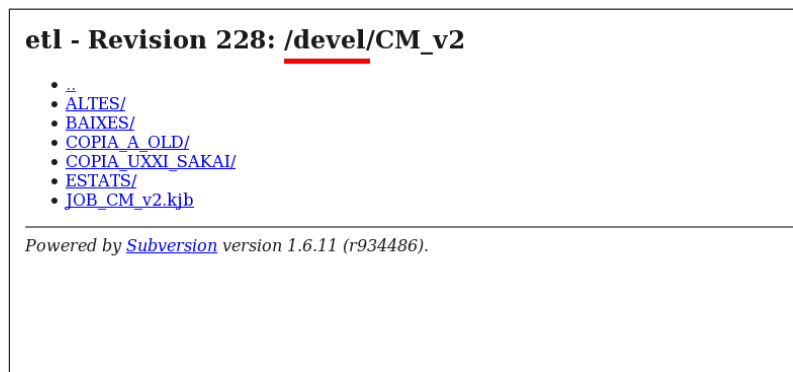


Figura 6.12: Repositori Subversion on desem ETLs en entorn devel

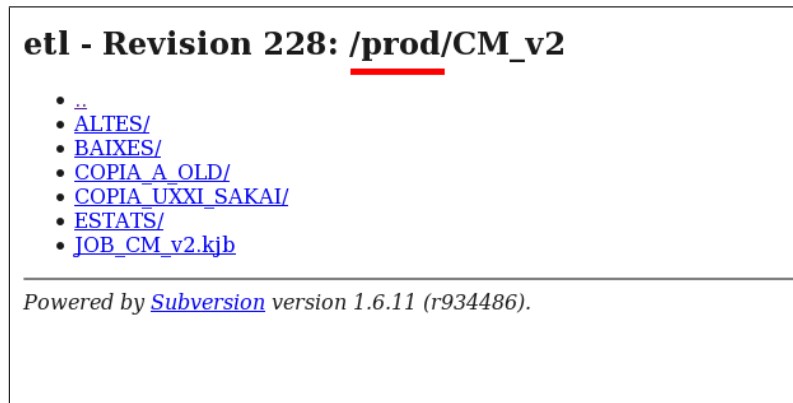


Figura 6.13: Repositori Subversion on desem ETLs en entorn prod

Per finalitzar el capítol es mostra gràficament l'ús d'aquestes eines en el nostre dia a dia i, en concret el aquest projecte:

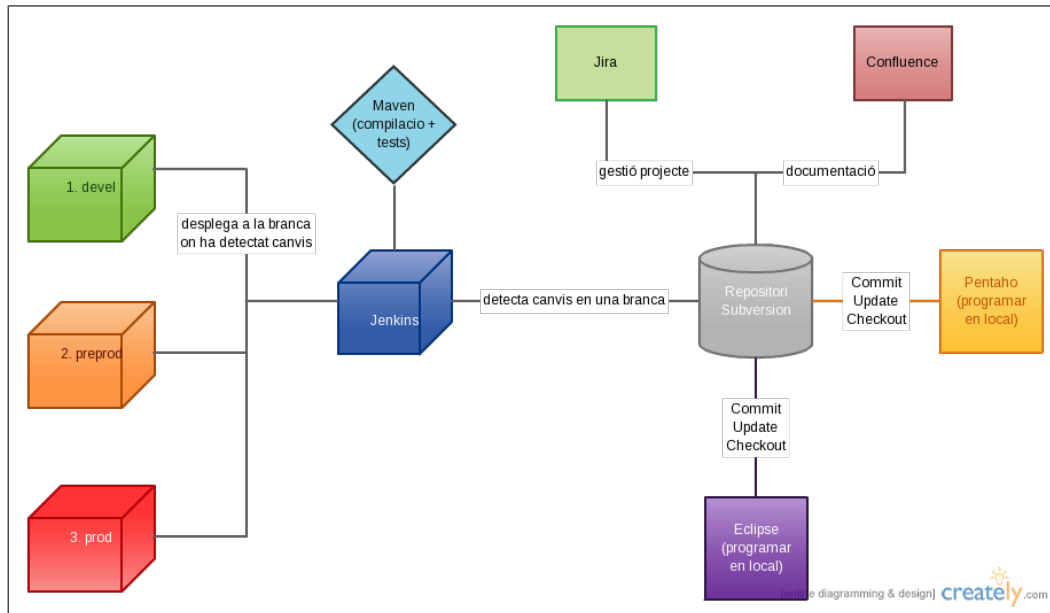


Figura 6.14: Ús eines en el desenvolupament

Capítol 7

Conclusions i treballs futurs

L'objectiu fonamental d'aquest projecte ha estat obtenir un model més obert i flexible que permeti al professorat gestionar la seva docència d'una forma més adaptada a les seves necessitats i preferències.

La idea ha estat evolucionar la integració entre Sakai i els sistemes origen de les dades, fent-la més completa, i intentant aconseguir alhora un millor procés d'integració i sincronització per tal de simplificar les tasques de l'equip que gestiona el Campus Virtual.

Aquest doble objectiu s'ha assolit amb èxit, doncs actualment i des de l'inici del curs acadèmic 2013-14 oferim als usuaris, però sobretot al professorat un Campus Virtual molt més obert i flexible. Avui dia els professors poden gaudir de la gestió dels seus espais de docència customitzant-los tant com vulguin. Poden també crear-se nous espais ja siguin col·laboratius, o de docència, i tot això fent ús de totes les dades que tenim sincronitzades amb el sistema origen (UXXI), de manera que poden gestionar directament tots els seus grups d'alumnes.

D'altra banda, tot i que la integració del Campus Virtual amb el sistema origen de les dades, continua essent un problema d'alta complexitat i variabilitat, actualment s'ha aconseguit implementar un procés molt ben estructurat i securitzat, que ens dona la tranquil·litat a l'equip que ho gestionem de saber que és un procés segur que no permet la pèrdua de dades, que no permet tampoc que ningú es quedi sense accés al Campus Virtual, però que a més, en el suposat cas que es produeixi un error en qualsevol dels passos, és capaç d'enviar un correu electrònic a tot l'equip avisant que s'ha produït un error i indicant en quin pas de tots els que realitza és el que ha fallat. De manera que, fins i tot en cas d'error, facilita la feina a l'equip acotant l'abast on cal mirar per resoldre-ho.

Com en tot projecte, ens hem trobat petits entrebancs que hem pogut anar solucionant. Destacaria potser, els problemes que ens vam trobar un cop acabada la primera versió del procés de sincronització. La idea que fos una solució el més simple possible ens va dur a fer uns plantejaments que van resultar insuficients en temes de seguretat. Com s'ha comentat en el capítol 5, ens vam adonar que el Campus Virtual quedava molt exposat a qualsevol fallada de connexió amb les bases de dades i, a més, el fet de ser un procés seqüencial feia que qualsevol errada durant el procés deixés el Campus Virtual desproveït d'usuaris.

No obstant, es va treballar en una segona versió molt més acurada. A més, amb tot l'aprenentatge que ja portàvem es va poder desenvolupar prou ràpidament. La migració d'una versió a l'altra no va ser gens traumàtica.

A data d'avui el Campus Virtual porta funcionant integrat amb el Course Management tot el curs acadèmic actual. Aquest fet és un clar indicador de l'excel·lent resultat del projecte, del qual personalment em sento plenament satisfeta i orgullosa, doncs demostra la qualitat de la feina realitzada per part de tot l'equip que ho fa possible.

Possibles millores o línies de treball futur

Durant tot aquest curs, s'han anat detectant inquietuds per part del professorat i suggeriments per millorar o per incorporar al Campus Virtual relacionats amb aquest projecte.

D'altra banda, fent una mica de retrospectiva hi ha certs aspectes que es podrien millorar.

Aspectes a millorar

- Enviar mail d'avís en cas d'error en qualsevol pas de la sincronització. Aquesta era una funcionalitat que estava prevista, però que amb el desenvolupament de la versió 2 ja es va afegir.
- Caldria escriure tests unitaris que ens servien per poder provar les diferents funcionalitats bàsiques del codi.
- Algun tipus de test automàtic o control que ens permetes descobrir si hi ha canvis en les estructures de dades origen.
- En els treballs de sincronització de dades amb el sistema origen, en cas d'error en la còpia a la versió anterior o còpia de Uxxi a Sakai, s'envia un mail d'error. Podríem desar els registres que cal tractar en algun fitxer i adjuntar-lo amb el mail, de manera que amb aquest fitxer i la taula de registre de les accions realitzades sabríem quins registres no s'han tractat.

Possibles línies de treball futur

- En l'estudi que s'ha presentat sobre els espais susceptibles de ser integrats amb el Course Management, n'hem descartat molts per no estar ben definit el sistema origen de les dades. Per tant, un possible treball futur seria aconseguir la integració amb la resta de tipus d'espais, com és per exemple els espais de tipus centre.
- Integrar el sistema origen GREC (Grups de Recerca) al Campus Virtual mitjançant el Course Management, de manera que es podria dotar d'espais de comunicació / espais de treball col·laboratiu als diferents grups de recerca de la UdL de forma automàtica, i amb una gestió automàtica dels participants.
- Incloure la gestió de grups de pràctiques i laboratoris. Si en el sistema origen s'introduïssin les dades referents a aules i calendaris on es duren a terme les pràctiques es podria incorporar al Campus Virtual mitjançant el Course Management.
- De la mateixa manera que s'ha deixat preparat per a un possible ús per part dels professors els grups d'alumnes separats per titulacions, es podrien deixar també preparats els grups d'alumnes separats per curs.
- En la pàgina del Campus Virtual on el professorat veu tots els grups en els que està planificat i que per tant en pot disposar, es barregen d'una banda els grups de matrícula, els grups de titulació i els grups dels espais de titulació. Una possible millora podria ser reprogramar aquesta pàgina per separar tot aquest llistat de grups segons el seu tipus en pestanyes separades, de manera que seria més intuïtiu el seu ús.

Suggeriments

Durant aquest curs acadèmic ens han arribat els següents suggeriments:

- Canviar el nom del rol 'Professor Ajudant', d'una banda per que el nom no agrada, però també per que no volen tenir menys permisos que un 'Professor'.
- L'eina *Guia Docent*, gestionada actualment per l'ICE, no es mostra correctament en els nous espais de docència que es creen els professors. Això es degut a que en la creació automàtica dels espais de d'assignatura que creem des de l'equip del Campus Virtual a l'inici de curs, forcem a que l'identificador de l'espai sigui igual al codi de l'assignatura. L'eina de la Guia Docent consulta aquest identificador per a cercar la guia docent corresponent. Però en els casos en què els professors s'han creat un nou espai, l'identificador de l'espai és un seguit de números i lletres aleatòri, de manera que quan l'eina intenta buscar la guia docent corresponent a aquest codi d'assignatura, no el troba.
- Els alumnes de la UdL que cursen alguna assignatura d'Erasmus en alguna altra universitat, surten als llistats que proveeix el Course Management en les assignatures. S'ha demanat que aquest alumnes no apareguin ja que no cal tenir-los en compte ni per planificar grups de pràctiques ni tampoc per a qualificar-los.
- Actualment, la informació que ens arriba referent als professors responsables de les assignatures no és la correcta. Estem assignant als professors principals de cada grup, com a professors responsables de l'assignatura. Cal treballar per a que les dades des de l'origen siguin les correctes i poder assignar directament als CourseOfferings aquests professors responsables.

Apèndix A

Detall dels fitxers inclosos en la implementació

En aquest Annex s'hi inclou, a títol informatiu:

- L'estructura de fitxers resultants de la implementació de la API del Course Management.
- Detall dels diferents jobs de Quartz que hem implementat.
- Estructura organitzada de directoris per a les transformacions i jobs de Pentaho Spoon.

El codi de totes aquestes implementacions, s'inclou en el cd adjuntat al projecte.

A continuació es mostra en detall el contingut del projecte **cm-udl**:

98 APÈNDIX A. DETALL DELS FITXERS INCLOSOS EN LA IMPLEMENTACIÓ



Figura A.1: Contingut resultant dels fitxers de la implementació

S'observa que el projecte està dividit en tres subcarpetes principals:

- **hibernate:** Aquesta carpeta inclou d'una banda els fitxers hibernate que defineixen la relació entre les taules que es crearan a la base de dades i les entitats. I d'altra banda els fitxers java definint cadascuna de les classes.

- **hibernate-pack:** Aquest directori inclou el fitxer components.xml. En aquest fitxer s'hi defineixen les dependències dels components que fem servir.
- **impl:** Aquest directori inclou la implementació del servei en sí.

Tot seguit mostrem també el detall dels treball de Quartz implementats per a la gestió del projecte:



Figura A.2: Detall fitxers resultants de la implementació dels treballs de Quartz

Finalment, es detalla l'estructura de directoris que s'ha utilitzat per organitzar totes les transformacions i treballs realitzats amb Spoon per a la sincronització de les dades amb el sistema origen:

Es pot observar que en el directori arrel s'hi desa el treball principal (JOB_CM_v2.kjb).



Figura A.3: Estructura general de directoris dels fitxers ETLs

En el directori ALTES s'hi ha inclòs el job principal que crida totes les transformacions que tracten les altes (alumnes, professors, assignatures, grups, ...) i les transformacions pròpiament dites.

Anàlogament, en el directori BAIXES, s'hi inclou també el job principal i les transformacions que el conformen.

ALTES	BAIXES
1_ACADEMIC_SESSIONS.ktr	BAIXA_2_TITULACIONS.ktr
2_TITULACIONS.ktr	BAIXA_3 ASSOFE.ktr
3 ASSOFE.ktr	BAIXA_4 PLASSOFE.ktr
4 PLASSOFE.ktr	BAIXA_5_ES_GRUPS_MAT.ktr
5_GRUPS_MAT.ktr	BAIXA_5_ES_GRUPS_TIT.ktr
5_GRUPS_TA.ktr	BAIXA_5_GRUPS_MAT.ktr
5_GRUPS_TIT.ktr	BAIXA_5_GRUPS_TA.ktr
6_ALUMNES_MAT.ktr	BAIXA_5_GRUPS_TIT.ktr
6_ALUMNES_TIT.ktr	BAIXA_6_ALUMNES_MAT.ktr
7_PROFESSORS_MAT.ktr	BAIXA_6_ALUMNES_TIT.ktr
7_PROFESSORS_RESPONSABLES.ktr	BAIXA_7_PROFESSORS_MAT.ktr
7_PROFESSORS_TA.ktr	BAIXA_7_PROFESSORS_RESPONSABLES.ktr
7_PROFESSORS_TIT.ktr	BAIXA_7_PROFESSORS_TA.ktr
10_COORDINADORS_ET.ktr	BAIXA_7_PROFESSORS_TIT.ktr
11_PROFESSORS_ET.ktr	BAIXA_10_COORDINADORS_ET.ktr
12_ESTUDIANTS_ET.ktr	BAIXA_11_PROFESSORS_ET.ktr
13_GRUPS_ET.ktr	BAIXA_12_ESTUDIANTS_ET.ktr
14_OFERTA_ET.ktr	BAIXA_13_GRUPS_ET.ktr
15_TITULACIONS_ET.ktr	BAIXA_14_OFERTA_ET.ktr
J_INSERTS_UPDATES.kjb	BAIXA_15_TITULACIONS_ET.ktr
J_IU_ALUMNES.kjb	BAIXA_ACADEMIC_SESSIONS.ktr
J_IU_ET.kjb	BAIXA_USUARIS_GENERICS.ktr
J_IU_OFERTA.kjb	J_BAIXA_ALUMNES.kjb
J_IU_PROFESSORS.kjb	J_BAIXA_ET.kjb
USUARIS_GENERICS.ktr	J_BAIXA_OFERTA.kjb
	J_BAIXA_PROFESSORS.kjb
	J_DELETEES.kjb

Figura A.4: Contingut dels directoris d'ALTES i BAIXES

D'altra banda, organitzem dins un directori les transformacions que copien la informació del Course Management actual a una còpia per a poder ser comparada.

I finalment el directori COPIA_UXXI_SAKAI, conté tot el necessari per importar des del sistema origen totes les dades imprescindibles per a tractar i encabir en el nostre model de Course Management.

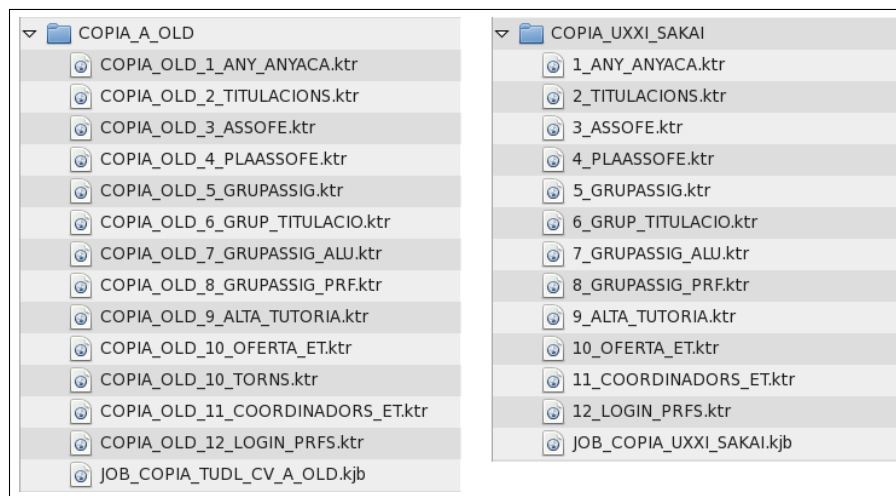


Figura A.5: Job i transformacions encarregades de fer una còpia de les dades actuals CourseManagement i de tornar a copiar les dades des del sistema origen

Per últim s'inclou un directori on desem les transformacions referents als tractaments d'estats de les diferents entitats que utilitzen els jobs de Quartz per a crear objectes al Campus Virtual (sites, sections, usuaris, ...) o bé per a enviar notificacions.

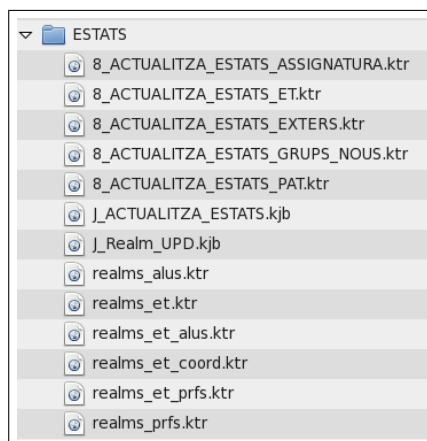


Figura A.6: Job i transformacions encarregades de tractar estats

Bibliografia

- [1] Estatuts de la Universitat de Lleida. Setembre 2003. DOGC.
- [2] EEES. Espacio Europeo Educacion Superior
www.eees.es
- [3] La nueva estructura de titulaciones
<http://www.educaweb.com/contenidos/educativos/espacio-europeo-educacion-superior-eees/nueva-estructura-titulaciones/>
- [4] Projecte Sakai
<http://sakaiproject.org/>
- [5] Alan Mark Berg, Michael Koruska. Sakai Courseware Management: The Official Guide. June 2009. Packt.
- [6] Course Management Overview
<https://confluence.sakaiproject.org/display/CM/Overview>
- [7] Course Management Documentation
<https://confluence.sakaiproject.org/display/DOC/Course+Management+Service+Documentation>
- [8] Course Management Tutorial
<https://confluence.sakaiproject.org/display/SAKDEV/Course+Management+Integration>

- [9] Course Management Model
<https://confluence.sakaiproject.org/display/~markjnorton/Sakai+Course+Management+Model>
- [10] Subversion
<http://subversion.tigris.org/>
- [11] Jira i Confluence
<https://www.atlassian.com>
- [12] Eclipse
www.eclipse.org
- [13] Pentaho
<http://community.pentaho.com>
- [14] Roland Bouman and Jos van Dongen. Pentaho Solutions. Business Intelligence and Data Warehousing with Pentaho and MySQL. 2009. Wiley.
- [15] Manual d'usuari de Spoon.
<http://wiki.pentaho.com/display/EAIes/Manual+del+Usuario+de+Spoon>